Hybrid Discourse Modeling and Summarization for a Speech-to-Speech Translation System

Dissertation
zur Erlangung des Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
der Technischen Fakultät I der
Universität des Saarlandes

vorgelegt von

Jan Alexandersson

Dekan: **Prof. Dr. Philipp Slusallek**

Erstgutachter: **Prof. Dr. Dr. h.c. mult. Wolfgang Wahlster**
Zweitgutachter: **Prof. Dr. Dietrich Klakow**

Saarbrücken
2003

Promotionskolloquium am 18.12.2003 um 16:15.
DFKI, Saarbrücken, Saal Turing.

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form in anderen Prüfungsverfahren vorgelegt.

Saarbrücken, den 6. November 2003

Danksagung

Vermutlich ist mein Freund und früherer Nachbar Nils Dahlbäck an dieser Arbeit schuld. Er ist der Typ, der mich davon überzeugt hat in Linköping Informatik zu studieren und nicht irgendwas anderes irgendwo anders, weit weg vom Heimatstädtchen. Desweiteren überzeugten er und seine Kollegen Mats Wirén, Magnus "Eximanatorn" Merkel, Arne Jönsson and Lars Ahrenberg mich, dass Computerlinguistik viel mehr Spass macht als, z.B. Prologprogrammierung. Ich tat mein bestes um diese Richtung nicht einzuschlagen, und konzentrierte mich einige Zeit auf andere Dinge. Wie dem auch sei, Mats ist es gelungen, mir den Weg zu bereiten am DFKI in Saarbrücken eine Tür zu öffnen um mit Computerlinguistik richtig arbeiten zu dürfen. Vielen Dank!

Das erste halbe Jahr am DFKI besetzte ich einen kleinen Teil von Stephan Busemann's Büro. Hier genoss ich eine wierklich angenehme Zeit - tackar tackar! Hiermit bedanke ich mich bei Prof. Hans Uszkoreit dafür, dass er es mir ermöglichte in der IUI-Gruppe am DFKI weiter zu arbeiten.

Den entscheidenden Anstoss für die vorliegende Arbeit gab mir meine anfängliche Tätigkeit in der IUI-Gruppe von Prof. Dr. Wolfgang Wahlster. Norbert Reithinger bat mich der Dialoggruppe von VerbMobil anzuschliessen, der auch Elisbeth Maier angehörte - vielen Dank! Im Rückblick kann ich sagen, dass meine ersten VerbMobil-Jahre die besten Voraussetzungen boten und optimale Bedingungen schufen um Computerlinguistik wirklich zu lernen.

Durch die Jahre hindurch, hatte ich am DFKI die Gelegenheit mit vielen interessanten Menschen zu arbeiten, die aus diesem Lebensabschnitt eine gute Zeit für mich machten, und desweiteren in der einen oder anderen Weise einen Beitrag zu dieser Arbeit geleistet haben: Insbesondere Norbert Reithinger, Elisabeth Maier, Michael Kipp, Ralf Engel, Peter Poller, und Tilman Becker, aber auch Wolfgang Finkler, Anne Kilger, Özlem Senay, Amy Demeisi, Massimo Romanelli, Patrick Bauer, Markus Löckelt, Reinhard Karger, die KollegInnen in den Projekten VerbMobil und SmartKom,

. . . (habe ich jemanden vergessen?). Mein ganz specieller Dank gilt Stephan Lesch für seine Hilfe bei der Implementierung und Entwicklung von den Konfabulationen. Ich bin den Koautoren meiner wissenschaftlichen Veröffentlichungen sehr dankbar für die fruchtbare Zusammenarbeit. Meinen wärmsten Dank an Norbert Pfleger's nicht-endenden, kamikatze-ähnlichen Leistungen wärend der Enstehung dieser Arbeit. Melanie Siegel und Mark Seligman erklärten vieles über Japan und die japanische Sprache - domo! Walter Kasper anwortete mindestens auf eintausend Fragen betreffend den wahren Kern der Semantik betreffend. Ein weiterer Dank an Tony Jameson für Konfabulationen und Korrekturlesen. Auch Amy Demeisi, Aude Wagnière und Tilman Becker danke ich für ihre unschätzbare Hilfe beim Korrekturlesen.

Tilman Becker, Uli Krieger, Frederik Fouvry, Takashi Ninomiya und Bernd Kiefer trugen wesentlich zum Verständniss der default-Unifikation wärend der Formalisierung der Overlay-Operation bei. Klaus Zechner sei bedankt für die Erlaubnis seine Software benutzen zu dürfen, obwohl er seine gestige und wissenschaftliche Heimat verlassen hat - Klaus: Ich dürfte sie nie benutzen. . . `:-(`.

Vielen Dank `www.leo.org` für die unschätzbare Übersetzungshilfe und `google` und `citeseer` für das Auffinden dessen was ich gesucht habe.

Einen besonderen Dank an Prof. Dr. Dr. h.c. mult. Wolfgang Wahlster für seinen unmissverständlichen Ansporn diese Arbeit endlich zu einem Ende zu bringen. Tack så mycket für den Tritt und dafür, dass Sie Ihre Zeit für Diskussionen, Vorschläge etc. investiert haben. Speziellen Dank gilt auch Prof. Dr. Dietrich Klakow, der kurzfristich die Aufgaben des Zweitgutachters übernommen hat.

Meiner Familie in Linköping ein warmes Dankeschön dafür, dass sie jederzeit einfach für mich da war und immer noch für mich da ist.

Zum Schluss, meine liebste Eva, ,,Marie-gumman" och ,,min Malin": Ohne eure Liebe, Geduld und Unterstützung und vorallem für die Aufmunterung als ich den Glauben in meinen Fähigkeiten verlor, wäre diese Arbeit nicht entstanden. (. . . und hier ist es nochmal:) Ich verspreche, dass ich *keine* weitere Dissertation schreiben werde.

Jan Alexandersson, November 2003

Acknowledgments

This work's fault is probably friend and former neighbor Nils Dahlbäcks! He is the guy who convinced me to study computer science in Linköping and not something else at some place far away from Linköping. Furthermore he and his colleagues Mats Wirén, Magnus "Eximanatorn" Merkel, Arne Jönsson and Lars Ahrenberg convinced me that NLP is much hipper than, say, logic programming. I tried my very best to avoid this direction and did for a while concentrate on other things. However, Mats finally managed to open up the door for really working with computational linguistics at DFKI in Saarbrücken. Thank you!

The first 6 months at DFKI I occupied a tiny part of Stephan Busemann's office. There, I enjoyed a very pleasant time - tackar, tackar. I hereby express my gratitude to Prof. Hans Uszkoreit for making it possible to continue working for the IUI group at DFKI.

The reason for this thesis started as I joined prof. Wolfgang Wahlster's IUI group. I was drafted by Norbert Reithinger to join the dialogue group of VerbMobil which additionally consisted of Elisbeth Maier - many thanks! Looking back, the first VerbMobil years formed the most optimal environment and the greatest opportunity to really learn computational linguistics ever seen!

During the years I came to work with a lot of people here at DFKI making this period of life a great time and furthermore contributing in one way or another to this thesis: Especially Norbert Reithinger, Elisabeth Maier, Michael Kipp, Ralf Engel, Peter Poller and Tilman Becker but also Martin Klesen, Wolfgang Finkler, Anne Kilger, Özlem Senay and Amy Demeisi, Massimo Romanelli, Patrick Bauer, Markus Löckelt, Reinhard Karger, the colleagues in the VerbMobil and SmartKom projects ... (did I forget someone?). A very special thank to Stephan Lesch for his help with implementation and development of the confabulations. I'm very grateful to my co-authors of the publications I've been involved in for fruitful collaboration. My warmest gratitude is due to Norbert Pfleger for his never-ending kamikatze-like efforts during the writing of this thesis. Melanie Siegel and Mark Seligman explained a lot about Japan and Japanese - domo! Walter Kasper answered at least one thousand questions about the true nature of semantics. Tony Jameson for confabulations. Amy Demeisi, Aude Wagnière and Tilman Becker for invaluable help with proofreading.

Tilman Becker, Uli Krieger, Frederik Fouvry, Takashi Ninomiya and Bernd Kiefer contributed enormously to my understanding of default unification during the formalization of the overlay operation. Thanks also to Klaus Zechner for allowing me to use his software although he left his home of dissertation - Klaus: I was never allowed to use it... `:-(`.

Thanks to `www.leo.org` for invaluable translation support and `google` and `citeseer` for finding most of the stuff I was looking for.

Very special thanks are due to Prof. Dr. Dr. h.c. mult. Wolfgang Wahlster for his irresistible encouragement to bring this thesis to its long-awaited completion. Tack så mycket for kicking me and for investing your time, for discussions, suggestions etc. I am also indebted to Prof. Dr. Dietrich Klakow for the “Zweitgutachten”.

I owe my family in Linköping a warm debt for just being there.

Finally, meine liebste Eva, “Marie-gumman” och “min Malin”, without your love, patience and support and, in particular, for the encouragement as I lost belief in my abilities, I would have never made it. (. . . and here it is again:) I promise I will *never ever* write another thesis.

Jan Alexandersson, November 2003

Warum entspricht uns die Welt nicht? Weil sie von selbst keinen Sinn hat. Wir sind so, daß wir etwas, was keinen Sinn hat, nicht ertragen. Also geben wir dem, was keinen Sinn hat, einen Sinn. Schon die Sinnlosigkeit zu erforschen, macht Sinn. Entspricht uns.

Why doesn't the world *reflect* us? Because it doesn't make sense by itself. We are such that we can't bear something that doesn't make sense. So we make sense of what doesn't make sense. Just investigating senselessness makes sense. *Reflects* us.

Martin Walser
"Sprache, sonst nichts"
"Die Zeit" No. 40, 30. September 1999

Kurzzusammenfassung

Die vorliegende Arbeit behandelt hauptsächlich zwei Themen, die für das VerbMobil-System, ein Übersetzungssystem gesprochener Spontansprache, entwickelt wurden: das Dialogmodell und als Applikation die multilinguale Generierung von Ergebnissprotokollen. Für die Dialogmodellierung sind zwei Themen von besonderem Interesse. Das erste behandelt eine in der vorliegenden Arbeit formalisierte Default-Unifikations-Operation namens Overlay, die als fundamentale Operation für Diskursverarbeitung dient. Das zweite besteht aus einem intentionalen Modell, das Intentionen eines Dialogs auf fünf Ebenen in einer sprachunabhängigen Repräsentation darstellt. Neben dem für die Protokollgenerierung entwickelten Generierungsalgorithmus wird eine umfassende Evaluation zur Protokollgenerierungsfunktionalität vorgestellt. Zusätzlich zu ,,precision“ und ,,recall“ wird ein neues Maß—Konfabulation (Engl.: ,,confabulation“)—vorgestellt, das eine präzisere Charakterisierung der Qualität eines komplexen Sprachverarbeitungssystems ermöglicht.

Short abstract

The thesis discusses two parts of the speech-to-speech translation system VerbMobil: the dialogue model and one of its applications, multilingual summary generation. In connection with the dialogue model, two topics are of special interest: (a) the use of a default unification operation called *overlay* as the fundamental operation for dialogue management; and (b) an intentional model that is able to describe intentions in dialogue on five levels in a language-independent way. Besides the actual generation algorithm developed, we present a comprehensive evaluation of the summarization functionality. In addition to precision and recall, a new characterization—confabulation—is defined that provides a more precise understanding of the performance of complex natural language processing systems.

Zusammenfassung

Die vorliegende Arbeit beschreibt die Ergebnisse mehrjähriger Forschung, die sich auf ein kleines Gebiet der Verarbeitung natürlicher Sprache konzentriert hat: ein robuster Ansatz zur Dialogmodellierung und Diskursverarbeitung mit dem übergeordneten Ziel von Sprachunabhängigkeit. Dieser Ansatz wurde hauptsächlich im Kontext von VerbMobil entwickelt—einem Projekt, das ein Übersetzungssystem für spontansprachliche Verhandlungsdialoge entwickelt hat.

Der erste wesentliche Beitrag dieser Arbeit ist ein Dialogmodell, das auf propositionalem Gehalt und Intentionen beruht. Zwei unterschiedliche Teile des Modells werden im Detail diskutiert: (a) Ein Ansatz zur Modellierung und Verfolgung des propositionalen Gehalts auf der Basis von Beschreibungslogiken und Defaultunifikation. Diese Technik wurde im Rahmen des SmartKom-Projekts weiterentwickelt, bei dem ein symetrisches multimodales Dialogsystem entwickelt wurde. Dies trifft insbesondere zu für die Weiterentwicklung und Formalisierung von *Überlagerung* (Engl.: “overlay”), einem Defaultunifikationsalgorithmus in Kombination mit einer Bewertungsfunktion der als Hauptverarbeitungsmechanismus für die Interpretation von Benutzerhypothesen dient. (b) Ein Ansatz für die Verfolgung von Intentionen, basierend auf Dialogakten, Dialogschritten und Dialogspielen kombiniert mit Dialogphasen. Diese Bausteine werden so arrangiert, dass ein gesamter Dialog sprachunabhängig auf fünf unterschiedlichen Ebenen beschrieben werden kann.

Im zweiten Hauptbeitrag dieser Arbeit wird eine auf dem Inhalt des Diskursgedächtnisses basierende Anwendung beschrieben: die multilinguale Generierung von Zusammenfassungen/Protokollen. Bei Verhandlungsdialogen enthält das Diskursgedächtnis am Ende des Dialogs unter den Diskursobjekten diejenigen Objekte, auf die die Teilnehmer sich verständigt haben. Basierend auf diesen wird ein datengetriebener ”bottom-up” Generierungsalgorithmus vorgestellt, der zusammen mit zwei bereits existierenden Modulen des VerbMobil-Systems–dem Transfermodul und dem multimodalen Gener-

ator GECO–Zusammenfassungen in allen Sprachen des Systems erzeugt.

Der dritte wesentliche Beitrag dieser Arbeit ist die Evaluation der Zusammenfassungsfunktionalität von VerbMobil. Die umfassende Evaluation belegt die Validität unseres Ansatzes zur Dialogmodellierung. Darüber hinaus wird gezeigt, dass die üblichen Evaluationsmasse, die auf ,,precision” und ,,recall” basieren, nicht geeignet sind, die Qualität der Protokollfunktionalität zu messen. Eine wesentliche Erkenntnis ist, dass Verarbeitungsfehler in allen Schritten neue Diskursobjekte in das Diskursgedächtnis einfügen können, die gar nicht Teil des Dialogs waren. Diese Objekte—Konfabulationen (Engl.: confabulations) genannt—tauchen schliesslich in den Zusammenfassungs/Protokollen auf. Die Standardevalutationsmasse ,,precision” und ,,recall” basieren auf der Annahme, dass eine Teilmenge der Diskursobjekte, die tatsächlich von den (Dialog-)Teilnehmern erwähnt werden, korrekterweise oder fälschlicherweise ausgewählt werden. Mit diesen Massen alleine werden die konfabulativen Fehler des Systems nicht in der Evaluation erkennbar. Daher wird eine ausführlichere Evaluation beschrieben, die Konfabulationen von echt-positiven Diskursobjekten unterscheidet. Mithilfe zweier neuer Metriken—*relative* und *totale Konfabulation*—wird eine informativere und ehrlichere Charakterisierung der Systemleistung präsentiert.

In der abschliessenden Diskussion werden die Ergebnisse der Arbeit zusammengefasst und weitere Forschungsthemen vorgeschlagen. Schliesslich enthält der Anhang Ausschnitte des Annotationshandbuchs für Dialogakte, -schritte und -spiele zusammen mit einigen Charakterisierungen des verwendeten Korpus. Er enthält auch ,,traces“ für zwei Beispieldialoge in VerbMobil zusammen mit den zugehörigen Ergebnisprotokollen.

Abstract

This thesis describes the result of several years of research focusing on a small part of natural language processing: with an overall goal of language-independence, a robust approach to dialogue modeling and discourse processing. The main context in which the approach has been developed is VerbMobil—a project for a speech-to-speech translation system for spontaneously spoken negotiation dialogue.

The first major contribution of the thesis is a dialogue model based on propositional content and intentions. Two distinct parts of the model are discussed in depth: (a) An approach to the modeling and tracking of propositional content that is based on description logics and default unification. This technique has been developed further within the SmartKom project—a project for symmetric multimodal dialogue. This is true in particular for the advancement and formalization of *overlay*, a default unification algorithm in combination with a scoring function that together serve as the main operation for the contextual interpretation of user hypotheses. (b) An approach to the modeling and tracking of intentions that is based on dialogue acts, dialogue moves and dialogue games together with dialogue phases. These building blocks can be arranged in such a way that the complete dialogue is described on five different levels in a language-independent way. A new characterization called *dialogue moves* is introduced that encompasses several dialogue acts.

In the second major contribution of the thesis, an application based on the content of the discourse memory is described: multilingual generation of summaries. For negotiative dialogue, the discourse memory contains at the end of the dialogue amongst other discourse objects those objects that have been agreed upon by both interlocutors. On the basis of these, a data-driven bottom-up generation algorithm is described that together with two already existing modules of the VerbMobil system—the transfer module and the multilingual generator GECO—produces summaries in any language deployed by the system.

The third major contribution of the thesis is the evaluation of the summarization functionality of VerbMobil. A comprehensive evaluation shows the validity of our approach to dialogue modeling. Additionally, it is shown that standard evaluation metrics based on precision and recall fail to describe the performance of the summarization functionality correctly. A crucial finding is that erroneous processing in any processing step inserts discourse objects that were not part of the dialogue into the discourse memory. These objects—called *confabulations*—eventually appear in the summaries. The standard evaluation metrics precision and recall are based on the assumption that a subset of those discourse objects that are actually mentioned by the interlocutors are correctly or erroneously selected. If these metrics are used, the number of confabulative errors committed by the system is never revealed. Therefore, a more extensive evaluation distinguishing confabulations from true positive discourse objects is described. Through the use of two new metrics—*relative* and *total* confabulation—a more honest characterization of the system performance is presented.

In the conclusion, we summarize the thesis and suggest further research directions. Finally, the appendix shows excerpts from the annotation manuals for dialogue acts, moves and games together with some corpus characteristics. It also shows traces of two sample dialogues processed by VerbMobil, along with the corresponding summaries.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contextual knowledge is necessary for understanding almost anything happening in the world; be it reading a sentence of a newspaper text or an article, listening to someone speaking or watching someone doing something. A computer system involved in some way in communication between, for example, two humans is no exception: it is essential for it to maintain a context for a correct interpretation or understanding of the observations made. This becomes evident in particular in short contributions. The knowledge that a question about the age of the dialogue participant has preceded the utterance *"forty-two"* makes the process of understanding an easy task. Another factor affecting the understanding is the *domain* in which the communication takes place. Knowledge about the domain often disambiguates the meaning of certain words, expressions and actions.

The way contextual knowledge is modeled, organized and managed in a computer system depends on the task performed by the computer system. Most systems have some notion of topic and focus or accessibility of referents. Simple task-oriented systems employ no discourse context at all. This is the case for most applications based on simple dialogue systems based on finite state technology, e. g., VoiceXML (see `http://www.voicexml.org/`). In such simple systems the interpretation of each word and expression is unambiguous and therefore there is no need for, e. g., clarifications, maybe except in the case where the speech recognizer fails. The drawback being that the dialogues in which the system can participate might be unnatural or artificial.

Engaged in a dialogue with another interlocutor, i. e., a human, a computer system can ask clarification questions to resolve ambiguities. This is not necessarily the case for a system mediating or eavesdropping a dialogue between two humans. The latter is the case for a translation system for spoken language like the VerbMobil system. The VerbMobil system is a speech-to-speech translation system for spontaneously spoken negotiation in two domains: appointment scheduling, travel planning (including hotel reservation and entertainments). In a different mode, the domain of PC maintenance is supported. The final VerbMobil system translates between English, German and Japanese[1]. Such a system needs a module that has the responsibility of maintaining contextual information for supporting translation.

Whereas the translation between languages within the same family (e. g., roman languages) is easier, more distant languages are harder to translate. One reason for this has to do with how the world is viewed in the cultures of the respective languages. For some cultures, there are situations or phenomena which can be described by fixed phrases or sometimes even a single word (see below). More related languages, like English and German, require less contextual information for an approximatively correct translation as, for example, German and Japanese. When translating to and from Japanese it is important to know, for example, who is speaking to whom and what social relation is between the speakers. The latter is important since, for instance, politeness is in Japanese a highly complicated task. Especially for a non-Japanese. This example is picked up by Levinson in his efforts in defining pragmatics. In (Levinson, 1983, page 10), he talks about language specific pragmatics:

> "...for example, the pragmatics of English might have relatively little to say about social status (beyond what we need to describe the appropriate contexts for the use of *sir, your honour* and the like), while in contrast the pragmatics of Japanese would be greatly concerned with the grammaticalization of the relative social ranks of the participants and referents."

In computer systems aimed for participating in a dialogue in some way, the contextual information is maintained in a part of the dialogue manager we call the *discourse memory* or *discourse manager.* It comprises data structures—which is used to record important parts of the dialogue—and algorithms used to update the content of the data structures or the *discourse state* for supporting the processing of other components in the system. One of the tasks of the discourse manager is to interpret sensory perception *in context.* These kind of interpretations are crucial for the functioning of the

[1] Additionally, Phillips and Siemens continued to integrate Mandarin into the system.

dialogue system. In man–machine dialogue systems, the discourse manager is often part of a module called the dialogue manager. The dialogue manager has the responsibility of initiating actions while confronted with stimuli. A typical action is the reaction on a user contribution by accessing an external resource, e. g., a database and then, depending on the outcome of the access, the initiatiation and possibly the generation of a response.

Contrary to such dialogue systems, in VerbMobil there exists no dialogue manager[2] in a traditional sense. This is since the VerbMobil system does not *control* the dialogue but merely *mediates* the dialogue. The reasons for this is multifarious but one of them is that experiences gained in so-called Wizard-of-Oz (WOZ) experiments (Dahlbäck, Jönsson, & Ahrenberg, 1993) showed, that a translation system intervening too often is not accepted by the users of the system (see chapter 3).

Despite its mediating role in VerbMobil, the dialogue manager has the task of keeping track of what has been uttered (content) and attitude towards the content. In case the content of the dialogue memory remains intact throughout the dialogue, it is possible, for instance at the end of the dialogue, to recapitulate or rephrase the dialogue. Additionally, by making use of the attitudes uttered towards the content of the utterances it is possible to construe the *result* of the negotiation. The user has then a document either affirming what was said and translated during the course of the dialogue, or functioning as a reminder.

The list of challenges and topics addressed by the VerbMobil project is very long, but we would like to highlight some particularly important and interesting ones by giving a short introduction of the state-of-the art and challenges:

Speech Recognition

At the time VerbMobil started in 1993, speaker dependent isolated word recognition with push-to-talk technology for very limited vocabulary size was deployed. During the course of the project, we witnessed an impressive development resulting in open-microphone continuous speaker-independent large vocabulary speech recognition. Some of the reasons for this were new modeling techniques, bigger and better corpora where enough of training material was available. Despite these advancements, the recognition rate is still far from perfect. In fact, the speaker independent, continuous speech

[2] Throughout the thesis we will use the terms "dialogue manager", "dialogue module" and "discourse manager" interchangeably for denoting the dialogue module of VerbMobil.

recognizers used in the final system have a word accuracy performance of between 76% and 89% (Waibel, Soltau, Schultz, Schaaf, & Metze, 2000).

Additionally, spoken language has no delimiters, like full stop or comma. Therefore, a system dealing with multi-sentence contributions has to split the input stream of words to chunks of words corresponding to what often are referred to as *utterances*. This process is called *segmentation*. However, the segmentation of a contribution is not always correctly performed.

Spontaneously Spoken Dialogue

Even though linguists and socio-linguists have been engaged in analyzing real-world situations, its formal descriptions in terms of, e. g., large grammars is unsolved. One of the unanswered questions before the VerbMobil project started was how people would behave while engaged in a negotiation interpreted by a machine instead of a human interpreter. A related example is that of, e. g., (Lakoff, 1973), where the use of so-called indirect speech acts (see section 2.3.1) are imposed by the "rules of politeness" saying that one should *be clear* and *be polite*. In case these rules give raise to a conflict, people tend to err on the side of politeness. But how does this affect the behaviour of a human being while talking to another human being via a machine?

One of the findings of this thesis is that people behave very differently using the VerbMobil system. This does not necessarily have to do with politeness, but rather other factors, like limited system performance compared to a human interpreter. Therefore, on the one hand, the structure of the multi-lingual dialogue itself as well as the contributions in the multi-lingual setting are simpler. On the other hand, most of the mono-lingual data in the VerbMobil corpus show more freedom in structure and language.

Machine Translation

Despite considerable efforts worldwide, at the project start, machine translation (MT) was (and still is) far from mature. In particular, the combination of MT and spontaneously spoken language seemed very challenging.

The general task of MT is to translate an expression from some source language (S) into one a target language (T). Within MT, one often refers to a diagram depicted in figure 1.1 called the "MT triangle" (Vauquois, 1975). There, "S" stands for *Source*, "T" for *Target* whereas "I" stands for *Interlingua*. The MT triangle is often used while characterizing an MT-system or an approach to MT in terms of, e. g., interlingua and transfer, or while discussing the tradeoffs involved in MT. In an ideal world, there is some language-neutral representation—interlingua—which can be used as intermediate representation. Translation would then consist of mapping the source onto the interlingua representation (analysis) and then map the interlingua representation onto the target representation (generation). Now, this theory is far from realizable in practice; in fact, the general consensus is that there will be no interlingua translation devices within the near future. This statement is based on the observation that even if a language analysis component is faced with a (restricted) domain which is known in advance, analysis competitions, e. g., MUC[3] often reach results in the area of 30–95% depending on the task. Instead, so-called transfer approaches are used where translation consists of mapping the source to the target possibly via some intermediate representation but where special knowledge about that particular *language pair* is used.

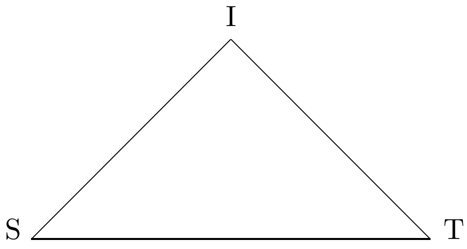


Figure 1.1: The Vauquois triangle

While it is easier (although not an easy task!) to translate between related languages, translating between, e. g., German and Japanese is more difficult. This has to do with a number of factors, like the different structure of the languages, honorifics, the way the world is described in terms of concepts, etc. European languages have of course, partly relatively small differences ranging from honorifics[4] to translation mismatches (see below).

Natural languages have different means of describing real-world entities by means of words and grammatical constructs. One language, or maybe better "culture" might use a single word for a certain real-world entity or action while another language might need several words. A simple example is the Swedish word *blunda* which translates to "eyes closed." The most

[3] MUC = Message Understating Conference.

[4] As I left, first person singular was always used to address a stranger in Sweden but this is never the case in German.

striking example is probably the Feugian (South American) word *Mamihlapinatapai* which translates to “to look at each other, each hoping the other will offer to do something which both parties much desire done but which neither is willing to do.” As we will see below, one has to have knowledge about context, and sometimes even a different view of how things are functioning to be able to correctly interpret and hence translate an expression.

The notions of *translation mismatches* and *translation divergences*, e. g., (Kameyama, Ochitani, & Peters, 1991a; Siegel, 1996), describe the phenomenon where an expression in the source language either does not contain enough information for a correct translation, or the target language can not express the meaning of the source. There are several classes of translation mismatches (see also (Kameyama, Ochitani, & Peters, 1991b)):

Number and definiteness In Japanese such information is often omitted. Example: *“hon wo yomimashita”* which can be translated to “I read {the,a,’ ’} book(s)”.

Perspective In Japanese, *Jibun* (self) can mean {my,your,him,...}self depending on context where, however, the reference can be far away in context. *jibun no accounto* means “account of *self.*”

Lexical translation mismatches Some words have no direct translation or are ambiguous. Some words mean something in between the concepts of related words. An example is the Japanese word *e* which means “painting” or “drawing” but not “photo;” it is more specific than “picture,” but more general than “painting” and “drawing.” The English word “go” corresponds, depending on context, to the German words “gehen” or “fahren.”

Speaker perspective In Japanese, honorifics plays an important role. Dependent on the relations between the speakers, a word like “give” has many different translations, e. g., *ageru, morau, kureru, kudusaru, sashiageru.* Therefore the translation from English to Japanese can be problematic.

These phenomena are not unique for the language pair Japanese – English. The Swedish utterance *jag väntar på min kompis* is ambiguous in the sense of speaker perspective. The default interpretation would read something like *I’m waiting for my buddy*, but the syntactic information allows for the reading *I’m waiting on my buddy.* In German, the buddy can be in accusative case—*Ich warte auf meinen Kumpel*—indicating a more situative meaning, or in dative—*Ich warte auf meinem Kumpel*—indicating that the speaker is physically located on his buddy waiting for someone/something else. Whereas humans develop strategies for dealing with these kind of ambiguities, an MT system translating this into German has to resolve the perspective.

A related phenomenon is that some sentences cannot be translated from one language to another using a one-to-one sentence approach. In some cases one needs more than one sentence in the target language to express the content of the source sentence correctly.

Another interesting characteristic of language hides behind the term *idioms*. They are not necessarily hard to translate. However, they need to be identified since the translation or maybe better analysis—if there is one—is not based on *compositionality* but on a fixed meaning.

Finally, there are other, more subtle challenges when dealing with language and translation which has to do with how language is used and how one communicates. My own experience (from being now 10 years abroad) is that in some *situations* there are no satisfying expressions in my mother tongue. There are translations or approximate expressions—yes—but they simply do not fit a 100%.

1.1 The VerbMobil Project

Prior to the actual project, a feasability study answering the question whether "VerbMobil was an appropriate goal to pursue." (Kay, Gawron, & Norvig, 1991, page 2) was conducted. The study went through the state of the art in the main research fields for the project—machine translation and speech recognition—with a fine-toothed comb. An important part of the study is the chapter containing the recommendations of how the project should be conducted.

The actual VerbMobil project was, at the time of writing, the largest European AI project (Wahlster, 2001). Starting in 1993, about 900 person years were spent in sub-projects ranging from speech recognition via analysis, transfer and generation to speech synthesis, corpus collection and annotation. Its first phase employed 125 persons per year for four years, whereas its second phase engaged about 100 persons per year over the next four years. At the end of the project in year 2000, a total of $80 million (private and public) funding was spend.

Besides the actual system (see below) the project produced a big corpus of both monolingual as well as bilingual dialogues. This is one of the reasons behind the success of the system: practically unlimited amounts of (annotated) data[5] for the training of statistical-based modules and functionalities. In total, 180 hours of dialogues were recorded (see (Karger & Wahlster, 2000) and the Bavarian Archive for Speech signals—BAS[6]). The corpus is available on CD-ROMs.

1.1.1 The VerbMobil scenario

Following other research projects, the VerbMobil project narrowed down the task of the system by means of a scenario. The basic setting was two business persons, negotiating a meeting possibly including a trip (to the meeting), accommodation and entertainments, e. g., having a dinner or visiting a theater.

An Example Dialogue

Throughout this thesis we will refer to a dialogue called "Reinhard4" as shown in figure 1.2. Although the dialogue is not taken from our corpus but is a constructed dialogue, we have chosen to use it since it contains most interesting phenomena which are of our concern. In particular, the summary generated for this dialogue (see chapter 4) contains more or less all information mentioned in the dialogue indicating that the discourse processing (see chapter 3) was successful.

One of our concerns in this thesis is that of generating summaries. In our view, a summary should contain the agreed-upon items of the dialogue. Such a (German) summary[7] for Reinhard4 is shown in figure 1.3.

A more complete version of Reinhard4 annotated with linguistic information as processed by the system is found in the appendix. There, also

[5] Some statisticians are probably protesting and I agree: Yes, for some tasks you cannot have enough data, but for the training of, e. g., the dialogue act recognition component there were enough!

[6] At the time of writing this thesis, their English home page is `http://www.phonetik.uni-muenchen.de/Bas/BasHomeeng.html`

[7] An English translation is: **Participants:** Speaker B, Thompson **Date:** 6.8.2002 **Time:** 3:12 pm to 3:13 pm **Theme:** Trip with accommodation and recreational activities *Result* **Scheduling:** Speaker B and Thompson agreed on a business meeting on the 20th of January 2002 at 11 am in Hanover. Speaker B and Thompson will meet on the 16th of January 2002 at 9:30 at the train station. **Traveling:** A trip was agreed upon. The outward journey to Hanover by train starts at 5 pm on the 19th of January 2002. **Accommodation:** A hotel in the city center was agreed upon. A single room costs 80 Euro. Thompson takes care of the reservation. **Entertainment:** A dinner at a restaurant was agreed upon. Thompson will reserve a table.

```
 G1a   sch"onen Guten Tag

 E2a   hello this is Thompson speaking
   b   hello hello Mr.  Schneider

 G3a   ja es geht um das Gesch"aftstreffen in Hannover
   b   das ist ja am zwanzigsten Januar um elf Uhr vormittags

 E4a   so we have to leave Munich at six o'clock

 G5a   vielleicht fahren wir lieber den Tag davor
   b   da gibt es einen Zug um zwei Uhr

 E6a   I would prefer to leave at five
   b   I am pretty busy that day

 G7a   ja gerne, k"onnen wir machen
   b   dann brauchen wir noch eine "ubernachtung

 E8a   yes

 G9a   ich kenne ein gutes Hotel im Stadtzentrum
   b   ein Einzelzimmer kostet achtzig Euro

E10a   that sounds pretty good
   b   can you please do the reservations

G11a   sicher dann machen wir ein gemeinsames Abendessen in einem Restaurant
   b   ich werde einen Tisch reservieren

E12a   that would be nice
   b   let us meet at the station on Wednesday

G13a   um halb zehn am Bahnhof

E14a   good see you then

G15a   bis dann
```

Figure 1.2: “Reinhard4”—The sample dialogue

the content of the discourse memory is found.

1.1.2 The VerbMobil system

One of the features of the VerbMobil project was that apart from the corpus, numerous publications, a book (Wahlster, 2000) (see also (Kay et al., 1991)) etc. the project produced a working system - the VerbMobil system (see figure 1.4). At the time of writing—three years after the end of the project—the system is still alive and can be tried out while visiting DFKI. The system is running on a SUN Ultra-Sparc 80 with 4 processors (450 MHz), 2 GB main memory, 8 GB swap, no special signal processing hardware, Desklab Gradient A/D converter or Sun internal audio device and close-speaking cordless microphones.

Figure 1.5 depicts the graphical user interface (GUI) of the system. With some exceptions, the buttons correspond to one or more modules contributing to language processing in some way. Between the buttons, there are paths indicating the main data flow between different modules. The bottom row buttons correspond to the modules for processing input and output to and from the system. Above this row, to the (lower) left, the different speech recognition buttons together with the prosody button ("prosodic analysis") are positioned. Above to the upper left, one finds the integrated processing and the different more linguistic oriented translation tracks ("semantic construction" etc). To the right of that, the deep linguistic translation track ("dialog semantics," "transfer" and "generation") is located. Below that, the focus of this thesis—hidden below the button "dialog and context evaluation"—is found. The shallow translation tracks—"statistical translation" and "case-based translation" are, as indicated by the data flow paths, processing the output directly from the prosody analysis. One important module—the selection module—has no button. Finally, to the lower right, the different synthesizers are found. Furthermore, the VerbMobil system actually consists of many more modules, e. g., technical ones responsible for message passing, logging etc (Klüter, Ndiaye, & Kirchmann, 2000).

The final system processes a vocabulary size of over 10.000 word forms for German, almost 7.000 for English and about 2.000 for Japanese. Obviously, the main focus has been on the two language pairs English–German and German–English and we will, in fact, focus on this language pair in this thesis.

One of the reasons of VerbMobil's success was its parallel translation tracks deploying different translation strategies all with their own characteristics when it comes to advantages and disadvantages. In (Wahlster, 2001),

```
                VERBMOBIL ERGEBNISPROTOKOLL Nr. 1
----------------------------------------------------------------------

Teilnehmer: Sprecher B, Thompson
Datum: 6.8.2002
Uhrzeit: 15:12 Uhr bis 15:13 Uhr
Thema: Reise mit Treffen Unterkunft und Freizeitgestaltung
----------------------------------------------------------------------

                        GESPR"ACHSERGEBNISSE:



Terminabsprache:
Sprecher B und Thompson vereinbarten ein Gesch"aftstreffen am 20.
Januar 2002 um 11 Uhr am Vormittag in Hannover. Sprecher B und
Thompson treffen sich am 16. Januar 2002 um halb 10 am Bahnhof.


Reiseplanung:
Eine Reise wurde vereinbart. Die Hinfahrt nach Hannover mit der Bahn
beginnt am 19. Januar 2002 um 5 Uhr am Nachmittag.


Unterkunft:
Ein Hotel im Stadtzentrum wurde vereinbart. Ein Einzelzimmer kostet
80 Euro. Thompson k"ummert sich um die Reservierung.


Freizeit:
Ein Essen in einem Restaurant wurde vereinbart. Thompson k"ummert sich
um die Reservierung.
----------------------------------------------------------------------
               Protokollgenerierung automatisch am 6.8.2002 15:15:58 h
```

Figure 1.3: Reinhard4 - The summary

Figure 1.4: A demonstration of the VerbMobil system.

the evaluation measure used in VerbMobil is described:

> “We call a translation “approximately correct”, if it preserves the intention of the speaker and the main information of his utterance.”

Given this, the multi-engine approach consisting of the translation tracks as described below, contributes to an average translation quality of 85%. The translation tracks are:

Example based translation Trained on the aligned bilingual VerbMobil corpus, the example based translation track (Auerswald, 2000) delivers a relatively good translation as long as the input matches the training data.

Dialogue-act based translation By robustly extracting the core intention in terms of dialogue act together with propositional content, this translation track (Reithinger & Engel, 2000) is robust against, in particular, speech recognition errors.

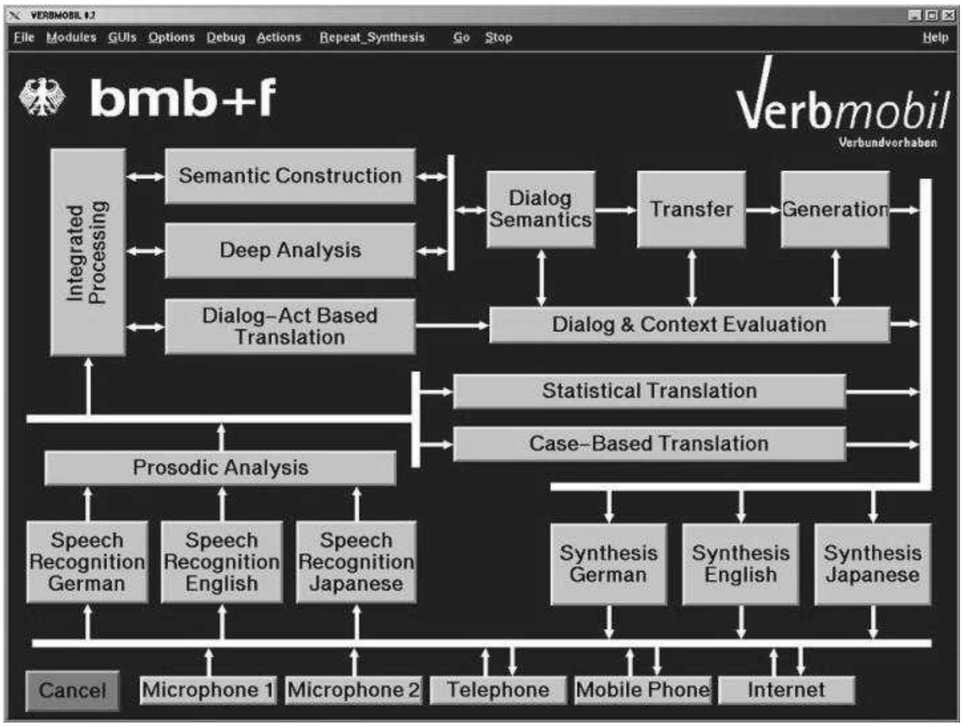


Figure 1.5: The VerbMobil graphical user interface (GUI)

Statistical translation Trained on the aligned bilingual VerbMobil corpus, the statistical translation track (Vogel et al., 2000) delivers a high approximate correctness especially for the language pair German–English.

Substring-based translation By combining statistical word alignment with precomputation of translation chunks together with contextual clustering, the substring based translation track (Block, Schachtl, & Gehrke, 2000) guarantees a translation with high approximate correctness.

Semantic-based translation Also known as the deep translation track, the semantic-based translation track deploys a classical computational linguistics approach using deep analysis (Schiehlen, 2000; Flickinger, Copestake, & Sag, 2000; Siegel, 2000; Rupp, Spilker, Klarner, & Worm, 2000; Pinkal, Rupp, & Worm, 2000; Bos & Heine, 2000), semantic transfer (Emele, Dorna, Lüdeling, Zinsmeister, & Rohrer, 2000) and generation (Becker, Kilger, Lopez, & Poller, 2000a). This track provides a high quality translation in case of success.

Next, we characterize the system by relating the different translation tracks to the translation triangle as described above. Depicted in figure 1.6, the different translation tracks are positioned according to to their functioning. Obviously, three translation tracks—example-based, statistical and substring-based—are rather *shallow.* The semantic or transfer based is more abstract, but is still far from denotable as interlingua. Actually, the dialogue act based translation is the translation track that is most interlingua-like. The reason for this characterization is that this track is based on two language independent information carriers, namely the dialogue act for representing the intention and the VerbMobil domain model for representing propositional content.

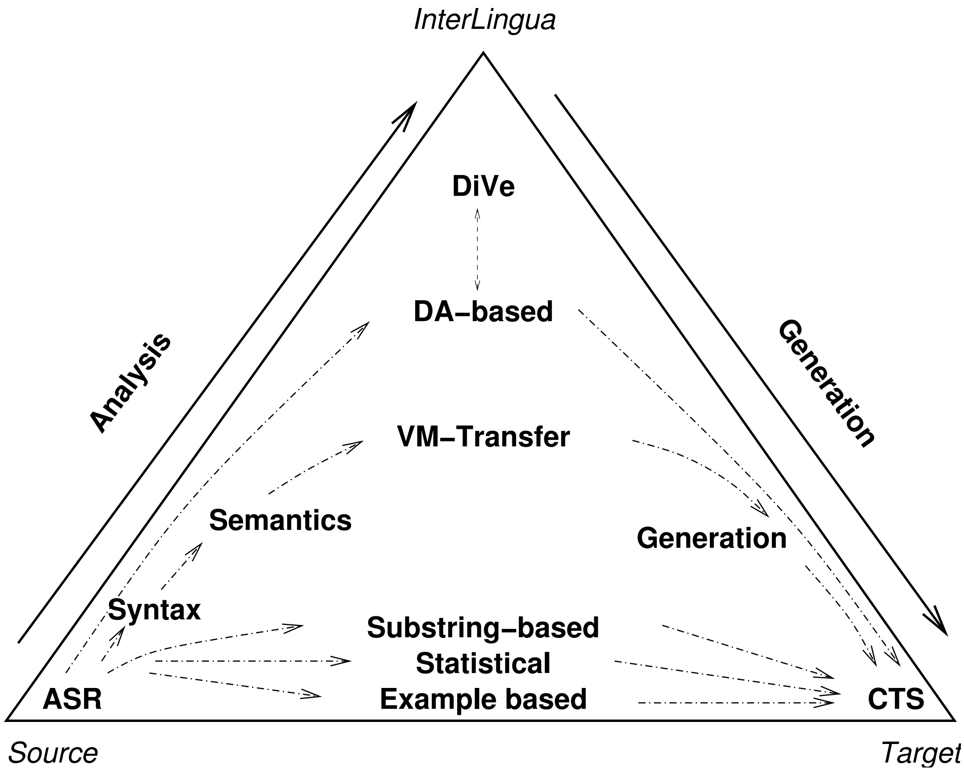


Figure 1.6: Relating VerbMobil to the Vauquois triangle.

The VerbMobil Interface Term - VIT

For the rest of the thesis we use a description language for linguistic information called VerbMobil *Interface Term* or short "VIT". The VIT is a uniform data structure serving as the linguistic information carrier mostly between the modules within the so-called deep processing track in VerbMobil. It encodes different linguistically motivated pieces of information, like (most notably) semantics, morpho-syntax, syntactic tense, semantic sorts, scope, prosody etc. But also the analyzed surface string and dialogue act is part of a VIT. As written in the introduction of (Dorna, 1996):

> "This information is linked to semantics and can be used for computing semantic tense, for disambiguation of under-specified analyses, for guiding semantic evaluation such as anaphora resolution, for adjacency or linear precedence determination, and for many more."

Figure 1.7 shows an example of the textual representation of the VIT representing the German sentence "*das ist wunderschön*" which translates approximately to *That is beautiful*—depending on context. Figure 1.8 shows a graphical representation of an English VIT.

1.2 Main Scientific Questions

Approaching the task of dialogue modeling in a speech–to–speech translation system for spontaneously spoken language, we put together a catalogue of unanswered research questions:

Representation issues How can we build up, maintain and exploit a representation of the dialogue in a way suitable for incremental processing and the support for translation and dialogue processing?

Controlling versus mediating What consequences for dialogue management are there due to the mediating role of VerbMobil?

Spontaneous speech What consequences on dialogue processing are there due to spontaneous speech?

Multilinguality What additional effects are there on dialogue processing due to the multilingual scenario?

Minutes and Summarization How can documents mirroring the result of the course of the negotiation be construed?

```
vit( vitID(sid(2,a,de,0,10,1,de,y,synger),         % SegmentID
           [word(das,1,[l28]),                      % WHG String
            word(ist,2,[l26]),
            word('wundersch"on',3,[l25])]),
     index(l24,l25,i7),                             % Index
     [stat(l25,i7),                                 % Conditions
      wunderschoen(l25,i8),
      decl(l24,h4),
      pron(l28,i8)],
     [leq(l23,h4),                                  % Constraints
      in_g(l26,l23),
      in_g(l28,l23),
      in_g(l25,l34)],
     [],                                            % Sorts
     [prontype(i8,third,demon)],                    % Discourse
     [gend(i8,neut),                                % Syntax
      num(i8,sg),
      pers(i8,3),
      cas(i8,nom)],
     [ta_mood(i7,ind),                              % Tense and Aspect
      ta_perf(i7,nonperf),
      ta_tense(i7,pres)],
     [pros_mood(l24,decl)]                          % Prosody
   )
```

Figure 1.7: Textual representation of the German VIT representing the sentence "*das ist wunderschön.*"

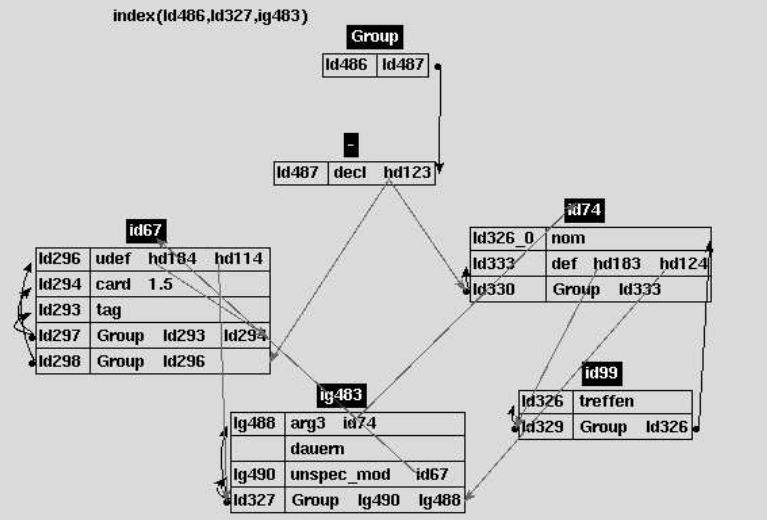


Figure 1.8: Graphical representation of a sample VIT. representing the sentence *Das Treffen dauert 1.5 Tage.* which translates to *The meeting lasts 1.5 days.*

Evaluation What metrics and methods can be used for evaluation? In addition to recognition and translation performance, we are particularly interested in the performance of the summarization functionality.

1.3 Thesis Organization

In the next chapter (chapter 2), we introduce some terminology followed by an overview over related and relevant research in the area of dialogue modeling. We are especially looking at the representation of utterances but also at relations between utterances and higher level structures. Before we put forward our own theory (sections 2.5 – 2.7) we briefly mention a way of verifying a theory by means of corpus annotation.

Chapter 3 starts with a look at some features of the VerbMobil corpus followed by a presentation of the performance of speech recognition and its effects (section 3.3). Thereafter, we summarize the main tasks and sketch the implementation of the dialogue module of VerbMobil in sections 3.4 and 3.6. The sections 3.7 and 3.8 contain a detailed description of the processing of the two main data structures of the dialogue module - the thematic structure and the intentional structure.

The third main pillar of the thesis—multilingual summary generation—is the content of chapter 4. There, we summarize relevant work in section 4.2 (summarization) and 4.3 (generation). In section 4.4, the summary generator is presented. Before we conclude the chapter, we evaluate our dialogue module by evaluating the summary generation functionality in section 4.5. Chapter 5 concludes the thesis and we point at some future directions.

Finally, the appendix contains two sample dialogues and their processing in the working system. Also, some linguistic information, e. g., the VerbMobil sortal ontology and some samples from our annotation manuals are given.

Chapter 2

Dialogue Modeling

The goal of this chapter is to lay the foundations for the dialogue modeling used by the dialogue component in VerbMobil. We start by surveying the relevant history and state-of-the-art in dialogue modeling. In particular we are interested in how we can represent chunks of communication, e. g., utterances, moves and turns in a language-independent way. We therefore consult work in the area of speech act theory and conversational games. It is important that we develop a theory which is tailored for the application to a working system processing spontaneous spoken language. Therefore, on the one hand, the model we finally end up with has to be fine grained enough to draw relevant inferences, but on the other hand it should not require that much detail that we will fail to use the model due to the processing circumstances, i. e., false recognition.

In section 2.3 we therefore identify models capable of contributing to a model of negotiation dialogues in the VerbMobil scenario. Section 2.4 surveys an important part of theories in general: corpus annotation, i. e., the process of, given a theory, annotating collected linguistic material with the theory. There we discuss more thoroughly, how we measure how well a corpus is annotated, and if we can be sure that, given the measurement, the theory is valid. Finally, we put forward our own model of negotiation dialogues in the VerbMobil scenario in sections 2.5–2.7. Whereas the first of these sections (2.5) has an introductory character, the second and third are concerned with two means for analyzing our negotiation dialogues. The former serves the purpose of characterizing a tree-like structure spanning over the whole dialogue. The goal of the latter is to contribute to, e. g., translation and the tracking of the objects being negotiated. Next, however, we give an introduction to this chapter with some historical remarks and some notes on the VerbMobil scenario (section 2.1). Section 2.2 introduces some of the terminology used, in particular, in this chapter but also throughout this thesis.

2.1 Introduction

Until the VerbMobil project started, most of the work within the NLP community concerned with the implementation of systems had focussed on dialogues between a human and a computer (henceforth man–machine dialogue). The first man–machine dialogue system was probably ELIZA (Weizenbaum, 1966). It is a computer program that communicates using natural language for playing the role of a psychiatrist. Among the fundamental issues dealt with, there is, e. g., *identification of key words* and *discovery of minimal context.* A milestone in the history of AI programs was SHRDLU, e. g., (Winograd, 1972). SHRDLU—developed in the late sixties[1]—is a computer program that controls a robot arm in a blocks world. The program understands and reacts on a wide variety of user contributions or speech acts (see below), including statements about, e. g., ownership of the items in the world. Additionally, it can answer questions about the reason for actions. Simple anaphorical expressions are resolved and questions concerning why a particular action was performed are answered. SHRDLU was one of the first systems that used *procedural semantics*, e. g., (Woods, 1981). Procedural semantics means that a certain natural language expression is associated with a piece of program code. Question-answering (Q/A) systems are systems assuming each input is one question and which produce an answer. One of the first Q/A systems was LUNAR (Woods & Kaplan, 1977). It was the first computer program that potentiated typing questions to a data base using natural language. LUNAR translated English questions about the chemical analysis of moon rocks into data base queries with a vocabulary site of about 3500 words. The syntactic analysis handled phenomena like tense, modality and even some anaphora and conjunctions. The processing of anaphora reveals the presence of some notion of discourse memory. Notable is that LUNAR is one of the first systems which was evaluated: Out of 111 questions, 78% were correctly answered. Later on, the TRAINS system, e. g., (Ferguson, Allen, Miller, & Ringger, 1996) allowed for spoken language. LINLIN (Jönsson, 1993) was an attempt to develop an typed interface to a travel agency. Both TRAINS and LINLIN belong to a category of dialogue systems which have to process ellipses and anaphor-

[1] See also `http://hci.stanford.edu/cs147/examples/shrdlu/`

ical expressions and hence require a more or less elaborate discourse model capable of resolving these kind of phenomena. One of the primary goals of these projects was to develop computer systems facilitating man–machine communication using written or spoken language. The modeling within these systems were aiming for the use, or direct implementation of dialogue systems. However, at some sites in Japan, Germany and USA, research on human–human models applied to, e. g., speech translation systems similar to that of the VerbMobil system has been carried out (Iida & Arita, 1992; Levin et al., 1995).

A wide range of models for describing certain aspects of dialogue or communication has been proposed in the past. Some of them are built on a quite simple dialogue structure, imposed by, for example, technical limitations of the system which the model is aimed at. Others, not necessarily more complicated models, have been developed from a more theoretical perspective. Within some of these models, sub-models characterise certain aspects of the dialogues, such as, clarification sub-dialogues. One of the reasons for the quite exhaustive modeling of clarification sub-dialogues are, for example, due to limitations of speech recognition, where, even today the best speaker independent speech recognizers perform quite badly (see section 3.3). Admittedly, for limited vocabulary and especially for speaker dependent recognition, current systems reach recognition accuracy in the region of 95% and more. Another good reason for the exhaustive modeling of clarification dialogues is when the dialogue system is not allowed to make any errors. This typically occurs when the dialogue system is involved in a critical task such as money transaction. For many systems dealing with spoken dialogue, short, single or two sentence user contributions were the basis of models and theories.

A common approach for processing user utterances is to represent them with speech acts together with the propositional content. Speech acts (or dialogue acts) can be viewed as labels mirroring the intention behind the utterance, e. g., posing a request (see section 2.3.1). The propositional content is us used to represent the semantic content of an utterance (without looking at the intention). The propositional content is commonly represented using, for instance, some logical form or frame-based formalisms (see section 2.7). Example of systems advocating the speech acts and propositional content approach are SHRDLU and TRAINS. There are, however, exceptions to this approach such as the one taken in the *sundial* project (McGlashan et al., 1992; Heisterkamp & McGlashan, 1996) where the result of the analysis is semantics as feature structures (SIL) only.

Aiming for the translation of spontaneous speech within a negotiation scenario, VerbMobil (Wahlster, 2000) is one of the first implemented broad coverage systems concerned with mediating human–human dialogues. The final system is able to translate between the three languages English, German and Japanese using a vocabulary of 10,000 for German, about 6,900 for English and around 2,500 for Japanese.[2] One of the efforts of the project was the large data collection (Karger & Wahlster, 2000) which resulted in 3,200 dialogs (German: 1,454; English: 726; Japanese: 1,020) of transcribed mono- as well as multi-lingual negotiation dialogues. Different parts of the corpus has been annotated with various linguistic information which has opened up the way for machine learning approaches.

The VerbMobil Scenario

The VerbMobil dialogues are negotiation dialogues between two business people, involved in the negotiation of a meeting time/place, travel planning, hotel booking and leisure time activities. In the VerbMobil setting no *barge-in*, i. e., the dialogue participant is interrupting the system during translation, is allowed. Instead, one interlocutor is speaking at a time and the contribution is then analyzed, translated, and verbalized in the target language by VerbMobil. After the translation has been synthesized, the next contribution is allowed to be uttered. One such contribution from one interlocutor is called a *turn*. A turn is possibly divided into *segments* which sometimes coincide with utterances or even sentences. In what follows, we will use the terms *segment* or *utterance* interchangeably to denote these parts of a turn. It is assumed, that each segment can be annotated with at least one dialogue act. A segment consists of at least one word as recognized by one of the speech recognizers or as transcribed in the VerbMobil corpus, see (Burger, 1997; Alexandersson et al., 1998). In the running system, it is assumed that each segment should be translated.

Our use of turn differs from that of, e. g., (Allwood, 1994) who uses turn to denote that a speaker is “holding the floor.” Using this definition it is thus allowed for another speaker to intervene with, for example, back-channeling utterances. Instead we have a more practical and technically oriented usage: it is not allowed for the listener to intervene during speaking or in the pause imposed by the system during processing.

Interestingly, human interpreters choose a translation strategy which is not based on a word-by-word translation or even sentence-by-sentence translation but is rather based on something we can be denoted *abstraction*

[2] Additionally, Siemens and Phillips has continued to integrate Mandarin into the system.

Original utterance Oh, Moment, ich glaube, Freitag habe ich einen festen Termin, da kann ich leider nicht. Also Freitags kann ich nicht, ich kann dienstags, mittwochs und donnerstags. Ham Sie da vielleicht noch einen Termin frei?

Literal translation Oops, one moment, I think Friday I have a regular appointment, unfortunately I can't then. So, Fridays I can't. I am free Tuesdays, Wednesdays and Thursdays. Are you free then?

Interpreter Translation Friday is impossible, but Tuesday, Wednesday, Thursday is okay.

Figure 2.1: Example of reduction

or *reduction* of the contribution. This is illustrated by the example in Figure 2.1. It shows a turn with its literal translation, and its translation by a human interpreter (Schmitz & Quantz, 1995).

The interpreter does not translate the individual segments but rather renders the intended interpretation of the turn. Worth noting is that this reduction of course eliminates performance phenomena like hesitations (*"oh"*) or repetitions (*"also freitags kann ich nicht"*), but also turn-giving segments (*"Ham Sie da vielleicht noch einen Termin frei?"*). The two core intentions of the contribution are, however, retained. The first part of the turn is *backward looking*; it refers back and responds to the previous proposal. The second part is concerned with displaying to the hearer a new, *forward looking* proposal. The interpreter chooses in this case not to translate the request (are you free then?) but instead a quite neutral translation. But also here the core intention is kept. Note, that such an approach to translation violates the basic assumption mentioned above that each segment should be translated.

Our corpus of negotiation dialogues are cluttered with such pairs of a forward looking (part of a) turn which is reacted upon in a backward looking (part of a) turn. Such patterns are well-known and described as *adjacency pairs*, *language games* and *conversational games* by several researches and philosophers, e. g., (Wittgenstein, 1974; Schegloff & Sacks, 1973; Kowtko, Isard, & Doherty, 1993).

It is possible to characterize our dialogues on other levels. For instance, our negotiation dialogues can be divided into *phases* such as *greeting* or *negotiation* phase. In most dialogues[3] the speakers start by greeting each other. Then, the topic of the dialogue might be mentioned[4] followed by the actual negotiation which can be divided into several negotiations. Finally, the result of the negotiation is sometimes concluded and there is some kind of farewell phase. A salient phenomenon is the difference in negotiation style: the American English part of the corpus shows more efficiency in that one speaker often verbalizes parts of his diary by posing several, alternative possibilities. The listener then just has to choose. The German speakers are more verbal and take their time, even asking the listener to suggest, for example, a date. For the Japanese part of the corpus other behaviour occurs, which is almost exotic to us. A suggestion not suitable for the listener is not directly rejected. Rather, it is commended, and rejected because of the adjacent suggestion from the listener (see also section 3.2, page 75).

2.2 Some terminology

In this section we summarize and discuss some terminology important for this thesis. Important is the definition of *pragmatics*. We slightly re-define pragmatics and, given this definition, we will show in the next chapter that we are in fact, with our work described here, able to at least partly carry out some of the challenges hidden behind this term.

2.2.1 Syntax, Semantics and Pragmatics

The trichotomy of syntax, semantics and pragmatics, originally introduced by Charles Morris (Morris, 1938), is one of the most popular ways of discriminating the study of human language or human language communication respectively. There are a wide variety of definitions of these concepts, especially since Morris' original definition of, for instance, pragmatics has split into several disciplines such as socio-linguistics, psycholinguistics, etc. Nevertheless we will recapitulate his original and some currently used definitions:

- **Syntax:**

[3] There are exceptions to almost every regularity described here.

[4] The topic is not negotiated - the instructions of the participants contains no alternative to the negotiation scenario. But some - I guess hungry or, maybe, bored subjects - try to solve the task as fast as possible and pose the first suggestion without greeting or considering the topic even being worth to mention.

In his classical work Morris describes syntax as "the study of the formal relation of signs to one another." A less formal description could be something like: syntax is used to denote the study of how words can be put together to form phrases, utterances or sentences in some natural language (cp. with syntax in formal languages).

- **Semantics:**

  According to Morris, semantics stands for "the study of the relation of signs to the objects to which the signs are applicable." Another view of semantics is "the study of situation independent meaning of utterances." To achieve this, one usually uses truth, reference and logical form.

- **Pragmatics:**

  Morris states: "the study of the relation of signs to interpreters." (Levinson, 1983) devotes a chapter to the discussion of the definition of the word pragmatics. One of these is: "pragmatics is the study of those principles that will account for why a certain set of sentences are anomalous, or not possible utterances." (Mey, 2001) uses the definition "Pragmatics studies the use of language in human communication as determined by the conditions of society." We argue that pragmatics stands for *The study of the principles for how non-anomalous communicative acts make sense.* Another possible definition of pragmatics is – contrary to the definition of semantics above – "pragmatics deals with the situation dependent meaning." We will return to our definition of pragmatics in chapter 3. There we describe a method for answering the question whether an utterance is pragmatically relevant or not given the state of a discourse memory.

Propositional Content

What is a proposition (*lat. prōpositiō*)? What is "propositional content?" (Bußmann, 1990; Honderich, 1995; Reber, 1996) present the following definitions:

- (Honderich, 1995, proposition (-al content)):

  > "The precise formulation varies, but a proposition, or propositional content, is customarily defined in modern logic as 'what is asserted' when a sentence (an indicative or declarative sentence) is used to say something true or false, or as

> 'what is expressed by' such a sentence. The term is also applied to what is expressed by the subordinate clauses of complex sentences, to forms of words which, if separated from the complex sentences of which they are part, can stand alone as indicative sentences in their own right. Accordingly, such sentences and clauses are often called 'propositional signs'. . . . "

- (Reber, 1996, proposition):

  > "4. A linguistic proposition is a formal statement that represents a component of the underlying meaning of a sentence. Here, the sentence 'apples are red' would be represented as (apple, all, red). The notion of truth here is irrelevant; the concern is with whether or not the proposition provides an accurate characterisation of the underlying meaning of the sentence being analysed."

- (Reber, 1996, propositional content)

  > "In linguistics, the full set of propositions (4) expressed by a sentence or paragraph or extended discourse."

- Hadumod Bußmann (Bußmann, 1990)

  According to Bussman, the term proposition has evolved out of the philosophy and logic. Later, the term has been adopted by the research fields linguistic semantics and speech act theory. In (Bußmann, 1990) her basis is on the work of, e. g., Russel, Austin and Searle. In her opinion, the proposition is:

  > der sprachunabhängige, bezüglich des Illokutionstyps neutrale gemeinsame Nenner der Bedeutung von Sätzen bezeichnet, die das Zutreffen eines Sachverhalts zum Inhalt haben.

  A possible translation[5] is *. . . the language independent meaning without taking the illocutionary force into account that assert the truth to the state of affairs under some circumstance.*

[5] A well-known online machine translation facility related to space fishes makes the following out of the quote: *languageindependent common denominators of the meaning of sentences, neutral concerning the Illokutionstyps, designates, which cover an applying of circumstances.*

To conclude, the propositional content of, for instance, a sentence seems to correspond with what is stated without looking at the illocutionary force (see section 2.3.1). In the following examples the *proposition* remains the same despite the different illocutionary force and sentence mode:

(1) Jan smiles casually

(2) Is is true that Jan smiles casually?

(3) Does Jan really casually smile?

A logical form could be something like `smiles_casually(jan)`.

2.3 Theories of Dialogue Structure

We discuss theories of dialogue structure relevant for this thesis or theories which one would think are relevant. This includes different aspects or levels of models and theories that describe primarily spontaneously spoken dialogue on different levels which can be used for constructing a model we will implement in the VerbMobil system. The requirements (see section 3.4) of our model are mainly posed by other modules or functionalities in the system, such as the transfer module or the generation of summaries or minutes.

2.3.1 Modeling utterances

For the modeling of utterances a vast variety of work is found in the literature. They almost all refer back to Austin and Searles' work on speech acts. These theories of speech acts try to map linguistic utterances onto a fixed alphabet of categories possibly in combination with, for instance, propositional content. As we will see below, a beloved child has many names – the categories bear names like *speech acts*, *dialog(ue) acts*, *communicative acts or actions* or *moves*. We start by walking through the important work concerning labels for utterances, continue with propositional content and finally, work concerned with higher level structures.

Speech acts

The theory of speech acts was first coined and formulated by Austin (Austin, 1962)[6] and a couple of years later by Searle (Searle, 1969). Austin showed

[6] Austin's classical paper was not written by himself! It was published posthumously by his students. It is thus questionable whether it really represents Austin's own view or

that the traditional semantics at that time almost completely neglected almost everything except the informative function of language as it appears in indicative statements. Instead he viewed communication – speaking as well as writing – as a kind of social action. Within his theory communication is built up from basic (atomic) items – speech acts – which distinguish three aspects or forces of an utterance or act:

- **locution** the utterance itself, its meaning and reference,
- **illocution** the conventional function of the utterance as it should be understood by the hearer(s),
- **perlocution** the consequence or effect of the audience.

Searle (Searle, 1969) continued the work of Austin. He claims that an utterance can be characterized as performing four different acts:

- **Uttering words** performing utterance acts
- **Referring and predicating** performing propositional acts
- **Stating, questioning, commanding, promising, etc.** performing illocutionary acts
- **Perlocutionary act** the consequence or effect – intended or not – achieved in an audience

In (Searle, 1975) he extends this approach and argues that his speech act theory accepts the basic principles of communication as defined by Grice (Grice, 1975). Presupposing the Gricean maxims, the speech act theory categorizes the acts a speaker performs into five general actions one can perform:

- **assertives** the speaker tells how things are (concluding, asserting)
- **directives** the speaker tries to get the listener(s) to do something (concluding, asserting)
- **commissives** the speaker commits to doing something (promising, threatening, offering)
- **expressives** the speaker expresses his attitude, feeling (thanking, apologising, welcoming, congratulating)

not, see (Allwood, 1977; Levinson, 1983).

- **declaratives** the speaker brings about changes (declaring war, christening, excommunicating, etc)

where each speech act is characterised by ten so-called *felicity conditions.*[7]

When introducing speech acts, one has to mention the term *indirect speech acts.* In sentences like "Can you pass me the salt?" the literal meaning is a question in which the hearer is asked whether she is able to pass the speaker the salt. However, the intention behind the utterance is more of a request which asks the hearer (in a polite way) to pass the salt, e. g., (Brown, 1980; Levinson, 1983; Gordon & Lakoff, 1975).

(Levinson, 1983) states that this list (assertives – declaratives) is "a disappointment in that it lacks a principled basis" and points at other classificatory schemes (Levinson, 1983, p. 241). He concludes (Levinson, 1983, p. 243 ff) that:

> (i) "all utterances not only serve to express propositions, but also to perform actions."
> (ii) "...there is one privileged level of action that can be called the illocutionary act – or, more simply, the speech act. ..."
> (iii) "...there is at least one utterance form of utterance (the explicit performance) that directly and conventionally expresses it[8] - namely, the explicit performative."
> (iv) "The proper characterization of illocutionary force is provided by specifying the set of felicity conditions for each force."

He continues to claim that

> "the combination of illocutionary force and the propositional content can be used to describe an utterance."

Important is that the illocutionary force constitutes something which cannot be captured by truth-conditional semantics. Rather, illocutionary acts are to be described in terms of felicity conditions, which are specifications for appropriate usage. The same goes for propositional content which can

[7] The "felicity conditions" were introduced by Austin as a tool for determining whether a performative succeeds or "goes wrong." The term "goes wrong" is used since performatives cannot be true or false: Suppose I utter "I hereby christen you, my daughter, Ida." Then it might be acceptable, if she has not been named something else. But if she already has, then the performative is not false but "went wrong" or is, in Austin's terminology, infelicitous.

[8] Any particular illocutionary force

receive “propositional content restrictions” which describe restrictions on the content together with particular illocutionary forces.

Also (Allwood, 1977) takes a critical look at the work by Austin and Searle. He points out that Austin’s definition is not exact enough: It is for instance unclear how one can distinguish between the intended and the achieved effect of an utterance in the mind of the hearer. Furthermore, the term “speech act” or more precisely the use of “speech” seems to restrict the speech act theory to spoken acts. But, human–human communication is not just spoken, but also non-verbal (see (Allwood, 1995a) for examples).

Communicative Acts

Among other critiques and suggestions, Allwood points out that the atomic view of communicative actions is suboptimal and, since communicative actions usually occur ”sequential in interaction” and not ”in isolation”, one might better study larger chunks of communication. Instead of speech acts, the term “communicative act” is suggested and a new “more suitable conceptual framework for the study of communicative actions” (Allwood, 1977) is put forward (see also (Allwood, 1976)). In the view of (Allwood, 1977), the following framework should be used for the study of communicative action:

1. The intention and purpose – intended effects – of a communicative action.

2. The actual overt behaviour used to perform the communicative action.

3. The context in which the communicative action is performed.

4. The actual achieved effects on a receiver of the act of communication (which does not necessarily coincide with the intended effect).

5. As an extension of 4, the notion of conventional force, i. e., the social consequences of a certain communicative action.

In this more general framework, we find in (1) the illocutionary force. (2) is a very general description of how the message is performed, e. g., language, gestures. Apart from (3) which is present in some form in most frameworks, Allwood uses Austin’s (and Searle1969’s) perlocutionary act in (4). The extension of (4) is to our knowledge not found in other frameworks.

Dialogue acts

The term *dialogue act* was coined by Harry Bunt. Dialogue act as a concept is part of the *Dynamic Interpretation Theory* (DIT) (e. g., (Bunt, 1994, 2000)) which consists of two main concepts: that of *context* and that of *dialogue acts*. A dialogue act can be viewed as a context-changing operator which, applied to a context, produces another, updated, context. We will not pursue the context part of DIT in too much detail, but briefly mention its aspects which are necessary for the text below. Context is described by five facets:

- **Linguistic context** is the (local) context surrounding, e. g., a word or a phrase.
- **The semantic context** is formed by the underlying task and the task domain, like objects, properties and relations relevant to the task.
- **The physical context** consists of global aspects like time and place plus local aspects like communicative channels etc.
- **The social context** means the type of interactive situation and the roles of the participants.
- **The cognitive context** comprises the participants beliefs, intentions, plans and other attitudes.

In DIT, dialogues are viewed (Bunt, 1994, p. 3)

> "in an action perspective and language as a means to perform context-changing actions."

He continues:

> "We have introduced the term '*dialogue act*' for referring to the *functional units used by the speaker to change the context*. These functional units do not correspond to natural language utterances in a simple way, because utterances in general are *multi-functional*, . . . "

DIT distinguishes between three aspects of an utterance or dialogue act:

- **The utterance form** of a dialogue act determines the changes to the linguistic context that a dialogue act causes.

- **The communicative function** defines precisely what significance the semantic content will have in the new context.
- **The semantic content** will have a particular significance in the new context (which it did not necessarily have before the dialogue act was performed).

As an example of the multi-functionality of dialogue acts, “thank you” in the context of an answer is used. It functions not only as an expression of gratitude, but also to inform that the answer was understood[9]. In addition – depending on intonation – the utterance may have a turn management function.

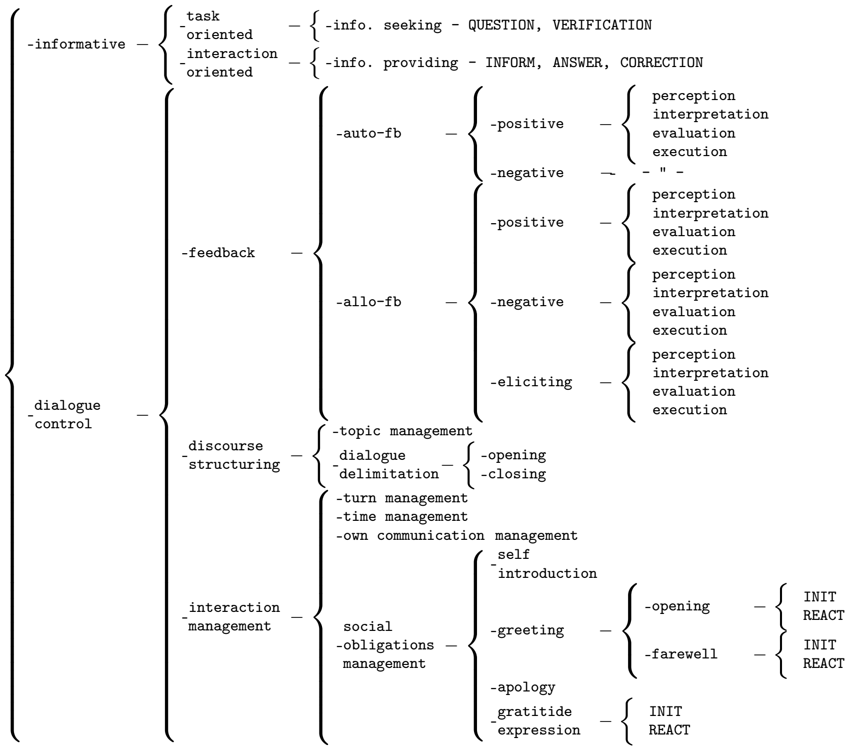


Figure 2.2: Harry Bunts Dialogue Control Functions

[9] Whether the answerer knows that the answer was correctly understood is left unanswered.

In (Bunt, 1994) the *dialogue control functions* are divided into two main parts. The first part covers `informative` dialogue functions. These are subdivided into `task-oriented` and `interaction oriented`. As examples of the former, `QUESTION` and `VERIFICATION` are given. Examples of the latter are `INFORM` and `ANSWER`. For the second part – dialogue control – three groupings are presented: `feedback`, `discourse structuring` and `interaction management`. Feedback functions provide the hearer with information about the speakers processing of the preceding utterance. Examples are `perception` and `evaluation`. `Topic management`, `opening` and `closing` are examples of the discourse structuring part. This grouping serves to indicate the speaker's view of the state of his/her plan for how to continue. Finally, the interaction management grouping consists of, e. g., `turn/time management` and `social obligation management` such as `apology` and different kinds of greetings: `welcome`. This grouping functions as a "smoothing glue" between turns and at the beginning and end of (parts of) dialogues. In (Bunt, 1996) the author adds the concept of *allo-feedback* together with some classes for "illustrative purposes." This add-on has been incorporated into Figure 2.2.

Clearly, and admittedly (personal communication), completeness of the inventory of dialogue control functions is problematic: the repertoire described above has been restricted to Dutch and English *information seeking* dialogues. Moreover, the system of dialogue control functions is "customizable." In subsequent papers, e. g., (Bunt, 1996, 1995, 2000), different aspects of the dialogue control function system are developed: For instance, in (Bunt, 1995), the author introduces the difference between dialogue control acts and task-oriented acts. The reason for differentiating dialogue acts is that they operate on different parts of the context. Whereas the task-oriented acts cause changes in the semantic context, the dialogue control acts may only cause changes in the social or physical context.

Dialogue Moves

The term *dialogue move* has been used in a number of publications including (Carlson, 1983; Larsson, 1998a). Within the TRINDI project (Larsson, 1998a) the term *move* is used. The reason for this is displayed in the first footnote of (Larsson, 1998a, p. 1):

> "There is a proliferation of names for this level of dialogue structure which perhaps is symptomatic of the lack of agreement between researchers. Apart from dialogue moves (Carlson) we

> have dialogue objects (Jönsson), communicative actions (Allen), communicative acts (Allwood and Traum), and the speech acts of Austin and Searle. Of course, there are also sometimes theoretical differences between many of these accounts. As noted, we will use the 'dialogue move' terminology which is derived from Wittgenstein's 'language game' metaphor, but this should not be taken as indicating any specific theoretical standpoint."

2.3.2 Relations between utterances and turns

Next, we turn to relations between utterances and turns. Relations like backward looking aspects are naturally found in mono- as well as dialogue since sentences or utterances are related to the previous context in some way or are unconnected, i. e., they introduce something new not previously mentioned. Important means related to communication management include (see (Allwood, 1994)) *basic communication feedback* (Allwood, Nivre, & Ahlsén, 1992), *turn management* (Sacks, Schegloff, & Jefferson, 1974), and *sequential structuring* (Schegloff & Sacks, 1973). The concept of forward/backward looking *center* is an important ingredient in the framework *centering* (Grosz, Joshi, & Weinstein, 1995).

Forward and Backward looking aspects

The concept of forward and backward looking aspects of an utterance has been used by a number of researchers, e. g., (Allwood, 1994, 1995a, 1995b; Larsson, 1998b) to characterize a certain aspect of the relation between communicative actions. Allwood argues that (Allwood, 1995b)

> "... every contribution in a dialogue, except possibly the first and the last, could be said to have both a backward looking reactive expressive aspect and a forward looking evocative aspect."

We argue that the first utterance can have a backward looking aspect as it might refer back to a dialogue prior to the current one. Also, the last contribution may consist of, for example, a promise to investigate something which might be the topic of another dialogue. In (Allwood, 1994, S. 7) the author writes:

> "With regard to forward and backward orientation, all parts of an utterance can be potentially relevant in both directions, however, the obligated functions are mainly backward-looking and the obligating functions are mainly forward-looking."

(Carletta, Dahlbäck, Reithinger, & Walker, 1997) describe the result of the Discourse Resource Initiative (DRI) meeting at Schloss Dagstuhl 1997. The goal of that meeting was to come up with a general framework for the annotation of different resources, i. e., corpora. Amongst other topics, two sub-groups discussed forward and backward looking aspects of utterances. Some of the aspects discussed in the forward looking group were, e. g., *Statement, Assert, Reassert*. Moreover, for the backward looking aspects of an utterance the four dimensions *understanding, agreement, information relations* and *answer*. In particular, the agreement dimension contains speakers attitude towards actions, plans objects etc. Succeeding work produced the DAMSL - Dialog Act Markup in Several Layers - scheme for the annotation of different aspects of, for example, forward/backward aspects of (parts of) dialogues, see (Allen & Core, 1997).

2.3.3 Initiative, Response and Mixed Initiative

Several researchers have looked into a phenomena which is known under the terms *initiative* and *response*. With the term *initiative* the modeling of "holding the floor"[10] (Sacks et al., 1974), "who is driving the dialogue" (Strayer & Heeman, 2001) or "controlling the dialogue" (Walker & Whittaker, 1990) is described. For mixed-initiative collaborative problem-solving dialogues (Smith, 1997) takes the view that

> "initiative lies with the participant whose goals currently have priority."

In (Guinn, 1996) a more global viewpoint is used:

> "An agent is said to have *dialogue initiative* over a mutual goal when that agent controls how that goal will be solved by the collaborators."

Also the term *mixed initiative* is used for the modeling of these aspects of dialogue, e. g., (Smith, 1997; Walker & Whittaker, 1990; Novick, 1988).

Initiativ och Respons

Inspired by the work of, for example, (Severinson-Eklund, 1983; Schegloff & Sacks, 1973; Goffman, 1981), Linell and Gustavsson (Linell & Gustavs-

[10] . . . in combination of turns, where the turn is a combination of the notions of *utterance*, *sentence* and *speech act*

son, 1987) describe a model called *"Initiativ och Respons"*[11] (henceforth IR) for dialogue communication that characterizes utterances into various types. The model is developed on the basis of recorded dialogues from various situations, such as court room negotiations, dinner chats, and class room lectures. The focus of their work lies on the dynamics and coherence of dialogue and on the dominance between the dialogue partners. Basic parts of the model are *initiative* and *response.* The authors use *initiative* for contributions where the speaker adds something new to the dialogue. *Response* is used for cases where the speaker refers back to the previous contribution. Thereby, the responses are defined in terms of how they refer to the context: focal, reference to own material or to the partner's contribution. Also whether the contribution is relevant to the previous discourse is distinguished.

The IR model is related to the work known under adjacency pairs, e. g., (Schegloff & Sacks, 1973) and "opening move" and "responding move" (Sinclair & Coulthard, 1975).

Initiative, Control and Mixed Initiative

One of the most cited work in the area of *mixed initiative* is (Walker & Whittaker, 1990) in which the authors discuss the two terms *mixed initiative* and *control* in *task oriented* and *advice giving*[12] dialogues. They explore how control is related to *attentional state.* The authors predict a shift of the attentional state where a shift in control is being performed. Due to the master-slave assumption for task-oriented dialogues, the control is expected to be held exclusively by the expert. On the other hand, they predict that the control is transfered back and forth within advice-giving dialogues.

The authors use four basic "utterance types:" *assertion, command, question* and *prompt* for the annotation of the dialogues. With these, rules for who has control of the dialogue are given:

Utterance Type	Controller
ASSERTION	SPEAKER (unless response to a question)
COMMAND	SPEAKER
QUESTION	SPEAKER (unless response to a question or command)
PROMPT	HEARER

[11] No folks, this is not a typo, it is genuine Swedish meaning—surprise, surprise—*Initiative and Response.* :-)

[12] The dialogues consist of ten financial support advice-giving dialogues (474 Turns) – see (Hirschberg & Litman, 1987) "Now lets talk about now", (Pollack, Hirschberg, & Webber, 1982) "User participation in the reasoning process of expert systems" and four dialogues (450 turns) (Whittaker & Stenton, 1988).

The thesis of their work is that *the controller corresponds to the Initiating Conversational Participant (ICP).*

In (Heeman & Strayer, 2001; Strayer & Heeman, 2001) the ideas put forward in (Walker & Whittaker, 1990) are applied to the TRAINS corpus; a part of the TRAINS corpus was annotated with the utterance types presented in (Walker & Whittaker, 1990). An inspection of the annotation indicated that the thesis of (Walker & Whittaker, 1990) is basically correct. However, it was found that the initiator of a discourse segment does not necessarily keep the initiative throughout the entire segment.

Initiative and Response on Two Levels

(Chu-Carroll & Brown, 1997, 1998) propose a two-level description of initiative-response which is able to distinguish between two different kinds of initiative. The authors argue that there is a difference in taking the initiative on the *dialogue level* and the *task level*, and that previous models fail to distinguish these. To clarify their issue they use the following example where a student (S) is consulting an advisor (A):

(1) *S* : *I want to take NLP to satisfy my seminar course requirement.*

(2) *Who is teaching NLP?*

(3*a*) *A* : *Dr. Smith is teaching NLP.*

(3*b*) *A* : *You can't take NLP this semester because you haven't taken AI, which is a prerequisite for the course.*

(3*c*) *A* : *You can't take NLP this semester because you haven't taken AI, which is a prerequisite for the course.*
I would suggest that you take some other seminar course to satisfy your requirement, and sign up as a listener for NLP if you're really interested in it.

(Chu-Carroll & Brown, 1997) argue that previous models, i. e., (Walker & Whittaker, 1990; Novick, 1988) can distinguish the difference between 3a and 3b (and between 3a and 3c) but fail to explain the difference between 3b and 3c. Using the wordings of (Chu-Carroll & Brown, 1997), "*in 3c A is more active*" in that she proposes an action which she believes should be incorporated into A's plan. An "agent" is said to have the *task initiative* if

> “she is directing how the agent’s task should be accomplished, i.e., if her utterances directly propose actions that the agents should perform”

Furthermore an agent is said to have the *dialogue initiative* if

> “she takes the conversational lead in order to establish mutual beliefs between the agents, such as mutual beliefs about a particular piece of domain knowledge or about the validity of a proposal”

Important here is the insight that dialogue initiative is subordinate in favour of task initiative:

> “Thus, when an agent takes over the task initiative, she also takes over the dialogue initiative, since the agent’s proposal of actions can be viewed as an attempt to establish the mutual belief that a set of actions be adopted by the agents.”

In (Chu-Carroll & Brown, 1998) their model is applied to a subset of the TRAINS91 corpus by annotating 16 dialogues with *task initiative holder* (THI) and *dialogue initiative holder* (DIH) obtaining the result showed in table 2.1.

Table 2.1: Distribution of task and dialogue initiative for a subset of the TRAINS91 corpus.

	TIH:System	TIH:User
DIH:System	37 (3.5%)	274 (26.3%)
DIH:User	4 (0.4%)	727 (69.8%)

Evidently in the majority of the turns, the task initiative holder holds the dialogue initiative. However, in almost 27 % of the turns, the

> “agents’ behaviour can be better accounted for by tracking the two types of initiatives separately.”

2.3.4 Conversational Games

One example of a corpus based on human–human dialogues is the map task corpus (Carletta et al., 1997a). In the dialogues recorded within this scenario, two humans are trying to duplicate a route drawn on one of the maps. However, the maps differ in details, and so the description of the route is not straight forward. There is sometimes eye contact, but the locutors are not allowed to see each other's map.

The modeling within the map task project is based on those of, e. g., (Sinclair & Coulthard, 1975; Carlson, 1983; Levin & Moore, 1977), but the conversational games model is described in (Kowtko et al., 1993). Three levels of structure have been developed and annotated. Starting with the lowest, *conversational moves* are the building blocks for the second level–*conversational games*. On top of these, on the third level the dialogues are divided into *transactions*.

Conversational moves are grouped into two different categories for modeling the dialogues:

- **initiating moves** INSTRUCT, EXPLAIN, CHECK, ALIGN, QUERY-YN, and QUERY-W

- **response moves** ACKNOWLEDGE, REPLY-W, and CLARIFY

Additionally, there is a third type of move - READY - which occur between two dialogues, and signals that the participant is prepared for a new dialogue.

Games are used to model the goal structure of a dialogue. It is composed by one initiating move and its succeeding moves until the goal of the initiating move has been fulfilled or abandoned. However, there are some naturally occurring aspects of spontaneous dialogue complicating this (theoretically nice) idea: It is not always clear for the hearer what goal the initiator of a game is trying to achieve. In general, a common phenomenon in spontaneous dialogues (which the map task dialogues prove) is embedded games which might serve new purposes not even salient in the current top level game. These might even be introduced while the dialogue partner is speaking. Therefore, the authors conclude that (Carletta et al., 1997, S. 3.4):

> "This makes it very difficult to develop a reliable coding scheme for complete game structures."

The annotated game structure can be viewed as a tree structure where the top level of each game is named by the purpose of the initiating move. Another view is that the conversational games, and indeed the transactions as well, just serve a bracketing purpose.

Typical for the dialogues in the map task corpus are the short turns (typically one or two moves). And even in the cases where the turns are longer they consist either of sequences of the same move type (typically sequences of INSTRUCT), or one ending of a game followed by one or two initiating moves. Another important observation is that the ending of an embedded game may also coincide with the ending of the conquering game.

2.3.5 Dialogue Games

In (Carlson, 1983), the fundamental unit for describing discourse is the so-called *language game*. It is based on the work by, e. g., Wittgenstein, Grice and Coulthard. Carlson writes

> "Wittgenstein did however not intend to develop his idea into a systematic theory of language games."

but continues to argue that there is

> "a very precise definition of a game (of strategy) in the mathematical theory of games developed by John von Neumann and Otto Morgenstein."

Within this theory there are three central concepts of *strategy*, *payoff* and *solution* to a game. These concepts are to be given for each participant of the game and a rational agent is one

> "who uses the most efficient means available to him to further his goals, i. e., one who follows his optimal strategies. The main virtue (and occasional weakness) of game and decision theory is its ability to explicate this key concept of goal-oriented action.

Carlson uses the following, rather strong characterization of "well-formedness of discourse":

> "A discourse is coherent if it can be extended into a well-formed dialogue game."

2.3.6 Plan Based Approaches

In the late seventies and early eighties, a plan based approach for modeling intentions for agents involved in cooperative dialogue was developed, e. g., (Cohen, 1978; Allen & Perrault, 1980). In particular, it was shown that by modeling the plans of an agent *and* another agent it was possible to provide a formal explanation of not just speech acts but also indirect speech acts and the reason for providing more information than asked for.

Within this line of research, an *agent* A is said to have *beliefs*[13], *intentions* and *goals*, e. g., acquire some information. To reach a goal he creates a *plan* which is then *executed*. The execution of the plan may be observable by another agent B. B may try *infer* the plan of A, a process which is often referred to as plan recognition, e. g., (Kautz, 1987). State-of-the-art problem solving systems, like STRIPS (Fikes & Nilsson, 1990), provides a formulation of actions and plans for these tasks.

Cohen (Cohen, 1978) showed that it is possible to model certain speech acts, like REQUEST and INFORM, actions in a planning system. Using the same planning paradigm, Allen and Perrault (Allen & Perrault, 1980) showed how three relevant tasks can be encoded in plans, namely

- generation of responses that provide more (but necessary) information than requested
- generation of responses providing an appropriate reaction to an utterance that consists solely of a sentence fragment
- generation of clarification dialogues for the case when the interpretation of an utterance is ambiguous

There is less or nothing written about the performance and evaluation of the approach. And, since these are highly knowledge intensive approaches, the robustness and coverage of such a system are questionable. One interesting design decision during the development of the TRAINS[14] system, e. g., (Allen et al., 1995), is revealed in (Allen, Miller, Ringger, & Sikorski, 1996) where the authors even admit that

> "We also knew that it would not be possible to directly use general models of plan recognition to aid in speech act interpretation (as in (Allen & Perrault, 1980; Litman & Allen, 1987; Carberry, 1990)), as these models would not lend themselves to real-time processing."

This argument is circumstantiated by the fact that there is no polynomial algorithm for STRIPS-like planners. However, the structure of plan based system is appealing, and the authors continue:

[13] Beliefs can be though of as knowledge, i. e., "things" the agent thinks he knows or are - true.

[14] See `http://www.cs.rochester.edu/research/trains/`

> "Our approach was to try to retain the overall structure of plan-based systems, but to use domain-specific reasoning techniques to provide real-time performance."

Using this domain-specific reasoning, in (Allen et al., 1996) an evaluation of the *complete* system is given.

Even today when the computers are 100 or maybe 1000 times faster than in the late seventies and in the early eighties, Allen admitted that (personal communication):

> "A straightforward implementation of those techniques as a heuristic search in a domain independent way would still be too inefficient—and too inaccurate—for a realistic application."

Still, a characteristic property of many of the dialogues in this line of research is the short user contributions. At most, a contribution consists of two utterances.

2.3.7 Plan Recognition Models

Contrary to the theoretical models presented above, (Iida & Arita, 1992) present a four layered plan recognition model which has been developed for a translation scenario. Although the material is monolingual (Japanese only) it shares a lot of the problems we have when developing models within the VerbMobil project. The model tries to describe a dialogue by means of a hierarchical tree structure, and consists of four different kind of plans:

- **Interaction plans** are connected to the utterances of the dialogue and allow for the exchange of information between speaker and hearer. Example of such a plan is a request–response pair (in their terminology: request–action unit).

- **Communication plans** roughly connect to the interaction plans and thus are an abstraction of the different possible realizations of the turn taking pair.

- **Dialogue plans** span a dialogue. The participants first opens up the dialogue, then talks about something (content), and finally closes the dialogue.

- **Domain plans** are the top level plan and allow for performing a given action as a sequence of other actions. As an example a registration for a conference is decomposed into "obtaining a registration form", "filling out the form", and finally "sending the form."

As evaluation corpus four dialogues are used. There is no quantitative evaluation but merely one example of the structure constructed by the plan recognizer. Furthermore, the model is constructed purely by hand.

Kautz suggests a formal plan recognition model based on events in a hierarchy (Kautz, 1987, 1991). His plans are more restrictive than those of, e. g., (Allen & Perrault, 1980). Still, his plan hierarchy has to be completed with circumscription. Kautz argues that the plans need to be formalized in such a restricted way since "...it would lead to massive increase in the size of the search space, since an infinite number of plans could be constructed by chaining on preconditions and effects." (Vilain, 1990) shows that plan recognition using this frame work can be translated into parsing with a context free grammar.

For more information on plan recognition, see (Kautz, 1991; Carberry, 1990).

2.3.8 Dialogue Grammars

Another line of research is the one concerned with describing dialogues with a (context free) grammar, e. g., (Scha & Polanyi, 1988; Jönsson, 1993). One of the arguments for this line is more pragmatic: some human—machine dialogues are simple enough to be described with a grammar and the use of, for instance, plan recognition models like in (Allen & Perrault, 1980).

LINLIN

Developing models for typed natural language interfaces, (Dahlbäck & Jönsson, 1992) describes a simple tree dialogue model—LINDA—which basically accepts IR-units. A move introducing a goal is called an initiative, and response if it is a goal-satisfying move. 98% of the 21 dialogues stemming from five different applications can be described by LINDA. 88% of same dialogues can be described by adjacency pairs. Figure 2.3 shows a sample dialogue grammar for an interface to a database with used cars taken from (Dahlbäck & Jönsson, 1992). There, Q_T and A_T are task related questions and answers and Q_D and A_D are reparation sequences initiated by the system.

SUNDIAL

In (Bilange, 1991), a task independent oral dialogue model for spoken dialogue developed within the SUNDIAL project is presented. SUNDIAL was

$$\begin{array}{lll}
D & ::= & Q_T/A_T\,|Q_X/A_S \\
IR & ::= & Q_T/A_T|Q_X/A_S \\
Q_T/A_T & ::= & Q_T(Q_D/A_D)^*(A_T) \\
Q_D/A_D & ::= & Q_D(A_D) \\
Q_X/A_S & ::= & Q_TA_S|Q_SA_S|Q_DA_S
\end{array}$$

Figure 2.3: A dialogue grammar for the Cars application. $*$, $+$, () and | have their usual meaning.

a European project whose goal was to develop generic oral cooperative dialogue systems. One if its main application was flight reservation. Basis for the model is two negotiation patterns:

1. Negotiation opening + Reaction
2. Negotiation opening + Reaction + Evaluation

Four decision layers are used:

Rules of conversation is divided into four levels:

- *Transition level* This level resemble a discourse segment (Grosz & Sidner, 1986).
- *Exchange level* This level models exchanges.
- *Intervention level* This level consists of the building blocks for the Exchange level. "Three possible illocutionary functions are attached to the interventions: *initiative*, *reaction* and *evaluation*."
- *Dialogue acts* In Addition to the core speech act, the dialogue act contains structural effects on the dialogue (Bunt, 1989).

System Dialogue Act Computation

User Dialogue Act interpretation

Strategic decision layer Since the model involves non-determinism, it is possible to continue the dialogue in several ways, this layer is concerned with selecting the next best action, i. e., dialogue act. A general strategy for oral dialogue is to avoid embedded structures.

The model gives raise to tree structures as depicted in figure 2.4 (Bilange, 1991).

$$T_1 \left[\begin{array}{l} \ldots \\ E_1 \left[\begin{array}{l} \mathit{initiative}(\mathit{system}\ldots) \\ \mathit{reaction} \\ \mathit{evaluation} : E_2 \left[\begin{array}{l} \mathit{initiative}(\mathit{system}\ldots) \\ \mathit{reaction}(\mathit{user}\ldots) \\ \mathit{evaluation}(\mathit{system}\ldots) \end{array}\right. \end{array}\right. \\ E_3 \left[\begin{array}{l} \mathit{initiative}(\mathit{system}\ldots) \\ \mathit{reaction}(\mathit{user}\ldots) \\ \mathit{evaluation}(\mathit{system}\ldots) \end{array}\right. \\ \ldots \end{array}\right.$$

Figure 2.4: Tree structure of an oral dialogue using the SUNDIAL dialogue manager.

2.3.9 DRT

The Discourse Representation Theory, e. g., (Kamp, 1981; Kamp & Reyle, 1993) is a semantic theory developed for the represention and the computation of

> "trans-sentential anaphora and other forms of text cohesion"

(Kamp & Scott, 1996). Other motivations for the development of DRT was donkey anaphora and tense and aspect. Although DRT was developed under a more theoretical perspective, there are now a lot of partial implementations running at several sites. In DRT, the fundamental structure for representing a discourse is the *discourse representation structure* (DRS) which consists of a set of *discourse referents* and set of *conditions* thereon. The set of discourse referents is the set of individuals of the discourse the DRS is representing. The different kinds of conditions are, e. g., predicates, negation and conditionals. The construction of a DRS consists of following a number of *construction rules*, e. g., the universal quantifier rule. The interpretation of a DRS consists of assigning the DRS a model: Given a DRS and a model, the DRS is said to be true iff there is a function f into the model which verifies the DRS.

2.3.10 Conclusion

The pioneer work by Austin and Searle resulted in an awareness, that truth semantics alone is not sufficient for modeling communication. Instead communication is composed by atomic items which received the name *speech* *acts*. The basic idea behind speech acts is present in a number of works, e. g., (Allwood, 1977; Bunt, 2000; Carlson, 1983; Larsson, 1998a) where, however, different aspects have received different levels of elaboration. The bottom line is that utterances should be modeled by at least something like the illocutionary force together with propositional content. This is exactly the approach taken in VerbMobil (see section 2.6.1).

For higher level structures, there are some similarities between the different models but what is modeled by the structures differ. There are approaches where communicative games can stand as representative. The idea is that there are certain obligations existent especially in dialogue, such as, one responses to a greeting with a greeting, answers a question, poses a question when one is requested to ask one.

Computationally, there are two major lines for modeling dialogue on an abstract level. The first is a computationally expensive approach based on STRIPS-like planning. This model is capable of explaining rather complex phenomena such as the reason for a, possibly indirect, speech act in a certain context (Allen & Perrault, 1980; Carberry, 1990; Litman & Allen, 1987). The other line follows the assumption that dialogue, both typed and oral, can be modeled by a (context free) grammar (Scha & Polanyi, 1988; Jönsson, 1993; Bilange, 1991; Kautz, 1987; Vilain, 1990). Research in the area of context free grammars has produced efficient (polynomial) algorithms for processing these grammars.

For a system like VerbMobil where the task is to follow the dialogue rather than control it and where real time processing is of great importance we adopt the latter approach (see section 2.6). Although regarded as too "ineloquent" (Allwood, 1995a) from a theoretical perspective, this choice is not necessarily uninteresting since it allows for a formal machinery expressive enough for drawing interesting conclusions, e. g., (Kautz, 1991; Vilain, 1990).

2.4 Using the Theory - Annotating Corpora

We now turn into the area of corpus annotation. There are huge efforts in both time and money world-wide invested in annotating corpora with different kinds of information, such as syntax, semantics, dialogue acts and conversation games. The reasons for annotating a corpus are manifold and amongst the ones which are prominent to us are the following:

- **Proof of concept** By annotating a corpus with a model, and especially if the annotation of several annotators are homogeneous, the annotation serves as a kind of proof of concept. However, unreliable coding does not necessarily indicate a non-perfect model. The annotation manual might be premature, and a revision of it might lead to better annotation results.

- **Supervised Learning** An annotated corpus can be used to gain experience not just by humans but by machines.

- **Evaluation** An annotated corpus can be used to evaluate the performance of a machine.

We will return to these topics later in this chapter and in chapter 3.

2.4.1 Coding Schemata—MATE

The MATE project (Klein et al., 1998) was, like DAMSL (Allen & Core, 1997), an effort to unify several dialogue act annotation schemata into one. Additionally an annotation tool was developed. The effort was based on the observation that there are a lot of similarities between the schemata.

Several groups have developed different annotation schemata for their corpus with their own set of dialogue acts or speech acts. All these groups were working on models for different tasks, like information retrieval/seeking dialogues (i. e., Alparon, and LinLin), analysis of child language (i. e., Chat), travel-planning (CSTAR), appointment-scheduling (JANUS, and VerbMobil), courtroom interaction (SLSA) and others. Some of these coding schemata or more precise *sets of dialogue acts* were additionally used in dialogue systems (i. e., JANUS, CSTAR, VerbMobil, and LinLin), whereas some were developed with more theoretical interests in mind (i. e., Chat, and SLSA). Almost all schemata investigated in the MATE project are task- and application-oriented contrary to more general domain-independence. The reason for this is that most of the schemata are geared towards some task-oriented application. Consequently these schemata are domain restricted too. Also, for most of the schemata several dialogues - in the range of 15 to some thousand - have been annotated. Most of the annotation schemata have been evaluated as well (see below).

Finally, the MATE project has judged three of these schemata to be “good” from a dialogue act perspective: Alparon, SWBD-DAMSL and VerbMobil. This judgement is based on the comparison in (Klein et al., 1998, p. 28–30), where the resources put into the annotation scheme as well as the annotation are listed. Also, the evaluation itself in the VerbMobil project is outstanding together with the efforts for SWBD-DAMSL.

2.4.2 Annotating Corpora reliably

We present and discuss the annotation of three relevant corpora: the map task corpus (Carletta et al., 1997a), the TRAINS91 corpus (Chu-Carroll & Brown, 1998), and the VerbMobil corpus (Maier, 1997; Reithinger & Kipp, 1998). To measure the quality of the annotation the inter-coder reliability κ (Cohen, 1960) is used:

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)} \tag{2.1}$$

In the formula 2.1 $P(A)$ represents the proportion of times that the coders agree, and $P(E)$ the proportion of times that one would expect them to agree by chance. Basis for the computation is a so-called confusion matrix which shows how the annotations coincide. Table 2.2 shows such a table for two annotators, annotating n dialogue acts on N utterances. There, a cell (n, m) where $n \neq m$ indicates that the annotators have different opinions about the annotation for a certain dialogue act, whereas the case $n = m$ reflect how often the annotators agree on a certain dialogue act.

Table 2.2: A confusion matrix for two annotators

	d_1	d_2	$\ldots$	d_n	
d_1	$f_{1,1}$	$f_{1,2}$	$\cdots$	$f_{1,n}$	$\sum f_{1,*}$
d_2	$f_{2,1}$	$f_{2,2}$	$\cdots$	$f_{2,n}$	$\sum f_{2,*}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	
d_n	$f_{n,1}$	$f_{n,2}$	$\cdots$	$f_{n,n}$	$\sum f_{n,*}$
	$\sum f_{*,1}$	$\sum f_{*,2}$	$\ldots$	$\sum f_{*,n}$	N

To compute $P(A)$, we sum the cases where the annotators agree divided through N:

$$P(A) = \frac{\sum_{i=1}^{n} f_{i,i}}{N} \tag{2.2}$$

and to compute $P(E)$:

$$P(E) = \frac{\sum_{i=1}^{n} f_{*,i} * f_{i,*}}{N^2} \tag{2.3}$$

There are different opinions about the usefulness or behaviour of the kappa statistics. Whereas some of the advantages of the kappa statistics are its ease in calculating and its appropriateness for testing whether agreement exceeds chance levels for binary and nominal ratings, the list of disadvantages is regrettably longer. Amongst the, to us, more relevant cons we have:

- Despite a high level of agreement and the fact that individual ratings are accurate, κ may be low. Whether a given kappa value implies a good or a bad rating system or diagnostic method depends on what model one assumes about the decision-making of raters (Uebersax, 1988).

- The κ is influenced by base-rates and trait prevalence. Consequently, kappa figures are seldom comparable across studies, procedures, or populations (Thompson & Walter, 1988; Feinstein & Cicchetti, 1990)."

Whereas the general opinion seems to be that a κ value greater than 0.8 indicates good agreement for, in our case, the annotators, a lower value does not necessarily indicate a lower agreement. For $0.67 < \kappa < 0.8$, tentative conclusions are allowed to be drawn (Carletta, 1996). However, for binary traits and binary ratings, especially where the data is skewed, e. g., 0.9/0.1 split of the data, a good agreement might result in a lower kappa value (Grove, Andreasen, McDonald-Scott, Keller, & Shapiro, 1981; Uebersax, 1987; Gwet, 2001). This observation implies that biases tend to move kappa down, rather than inflate it. A good κ value should be used as an indication for good agreement, but further analysis on absolute agreement using the proportions of specific positive/negative agreement might shed more light on the actual situation.

To conclude, the kappa statistics can be used for, e. g., evaluating intercoder agreement, but it should be interpreted with care. It could, on the other hand, be followed up with additional analysis. Recent suggestions advocate alternative measurements, e. g., the *AC1* statistics (Gwet, 2001).

We summarize some of the corpus annotations relevant to our work:

The Map Task Corpus (Carletta et al., 1997a) In what follows K stands for "number of coders", N for "Number of units". For move segmentation $\kappa = .92, N = 4097, K = 4$ was reached using word boundaries as units, and for move classification $\kappa = .83, N = 563, K = 4$. For game coding the agreement for where the games start and, for agreed starts, where they end has been used as metrics. Pairwise percent agreement is used. Agreement for the beginning of game reached 70% with an agreement of type of game of $\kappa = .86, N = 154, K = 4$. In (Carletta et al., 1997a) it is concluded that there is room for improvement of the annotation scheme. For the cross-coding agreement the problems arise where “the dialogue participants begin to overlap their utterances or fail to address each other’s concerns clearly.”

TIH-DIH (Chu-Carroll & Brown, 1998) an experiment with the annotation of task and dialogue initiative on 16 dialogues from the TRAINS-91 corpus. In total, three humans annotated “approximatively 10 %” of the corpus, which probably corresponds to one or maybe two dialogues. The inter-coder reliability was tested using the kappa measure obtaining $\kappa = 0.57$ for the task initiative holder and $\kappa = 0.69$ for the dialogue initiative holder, thus probably rendering the annotation irrelevant. To explain the reason for not getting better κ values, the authors points at the above-mentioned effect with κ in combination with skewed data. This is clearly the case (see table 2.1 on page 38). Another reason might be due to too few a number of annotated dialogues.

VERBMOBIL At two points during the project an evaluation of the coding of dialogue acts as well as the segmentation was performed (Maier, 1997). During the first phase of VERBMOBIL, the dialogue acts were defined in (Jekat et al., 1995), a document also serving as coding manual. The inter-coder reliability during the first phase was evaluated on 10 dialogues. For segmentation, given the boundary agreement $P(A) = 0.9849$, we have $\kappa = 0.9293$. For the dialogue acts, ten pre-segmented were annotated by two annotators giving $P(A) = 0.8294$. Therefore we have $\kappa = 0.7975$

During the second phase, the inter-coder reliability of dialogue acts was tested for ten dialogues (Reithinger & Kipp, 1998). Given $P(A) = 0.8530$ we have $\kappa = 0.8261$. One coder was asked to re-annotate 5 of the dialogues, obtaining $\kappa = 0.8430$ ($P(A) = 85.86\%$).

In total, 1505 dialogues (738 German, 375 English and 402 Japanese) have been annotated with dialogue acts. This corresponds to 76210 dialogue acts (37954 German, 22682 English and 15574 Japanese).

Conclusions

We make two important observations. The inter-coder reliability for the VERBMOBIL corpus is:

- comparable with that of another well known and widely accepted annotated corpus – the map task corpus – being aware that such a comparison is questionable, e. g., (Thompson & Walter, 1988; Feinstein & Cicchetti, 1990).

- good enough to show the validity of the definition of our dialogue acts as defined in (Alexandersson et al., 1998). This opens up the door for actually *using* our corpus for, e. g., machine learning methods (see section 3.5.1).

2.5 Dialogue Modeling in VerbMobil

We are now in the position of describing the models used by the dialogue module in VerbMobil. At this point we would like to stress that VerbMobil is a translation system. Its purpose is to translate utterances from one language to another. The translation is performed within a negotiation scenario. Whereas the task for a man–machine dialogue system is to act as a dialogue partner, the tasks for a dialogue component within a translation scenario is to *track* the dialogue[15]. Our processing (see chapter 3) is therefore focussed on providing contextual information which can be used to improve the overall translation:

- **Prediction** What is going to happened next?
- **Recognition** What is the most reasonable interpretation of a certain sign?
- **Translation** What is the most reasonable disambiguation of a sign, e. g., word or phrase?
- **Generation** (How) has an negotiation object been uttered/realized?

To reach this goal we have developed two means of representing the dialogues. The first one, called the *thematic structure*, is concerned with representing the content and status of the negotiation. This is modeled and represented by the propositional content (see section 2.2.1) together with the dialogue act. The information contained in this structure can be used to, for instance, generate summaries, but can contribute to realizing the ideas presented in (Schmitz & Quantz, 1995), where the authors point at the *redundancy* phenomena common in spontaneous speech (see Figure 2.1). The second structure we call the *intentional structure*. This tree

[15] see section 3.7.4 for a discussion of the similarities between the discourse processing in a mediating system (VerbMobil) and a man–machine system (SmartKom)

structure is used to represent different abstractions of the dialogue, e. g., *the dialogue phase*. As it turns out, this information can be used to improve the translation of words and phrases (Alexandersson et al., 1998).

The model presented below - the intentional structure including, e. g., dialogue acts and dialogue games, as well as the design of the propositional content - is based on our rather large corpus analysis effort.

In the next section (2.6) we will start by introducing the intentional structure. The propositional content is presented in section 2.7.

2.6 The Intentional Structure

Our main objective within VerbMobil with the intentional structure are twofold: First, it is used to support the translation process (Alexandersson et al., 1998). Some utterances, phrases, and words like the German phrase "Guten Tag" could be translated to "Hello" or "Good bye" depending on the *dialogue phase* (see Section 2.6.3 and 2.6.4). Second, it has the potential to be used to form the basis of the generation of minutes (see chapter 3). However, the exercise of designing a model capable of describing such a structure requires influences from much previous work as well as novel ideas. Especially when it comes to building the structure in the running system (see chapter 3), the use of techniques and research results from the area of probabilistic language modeling and machine learning has turned out to be fruitful.

The design of the intentional structure is depicted in Figure 2.5. It is a hierarchical structure composed of 5 *levels*, where each level can be viewed as a more abstract representation of the dialogue the higher up in the tree we go. A basic assumption is that cooperative negotiation dialogues can be represented by a tree structure. Much of the work presented below resembles work described elsewhere, e. g., (Bunt, 2000; Chu-Carroll & Brown, 1998; Carletta, 1996; Allwood, 2000; Levinson, 1983) but some parts, e.g., moves are, to our knowledge, novel. The levels are (from the bottom):

- **dialogue act** level. This level is an abstraction of the utterances[16]. The dialogue act received its name because the ideas presented within the *Dynamic Interpretation Theory*, e. g., (Bunt, 1996) resemble much of the modeling in VerbMobil. But also the description in (Levinson, 1983) has inspired our definition. Our dialogue acts are discussed in section 2.6.1.

[16] We will also use the more technical term "segments" in the next chapter.

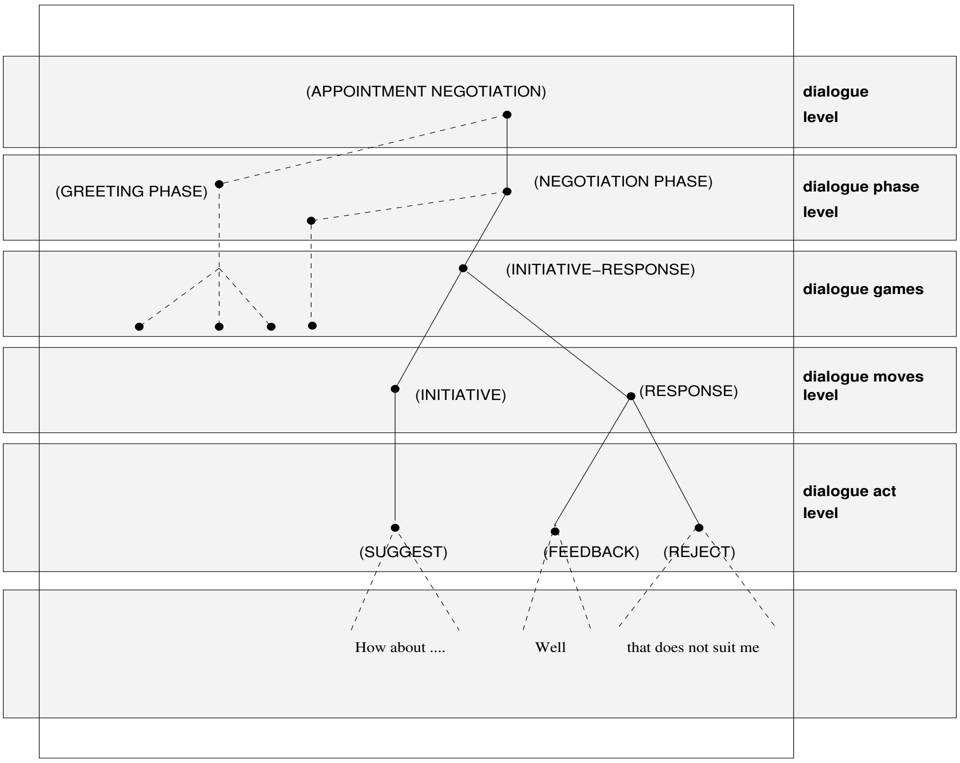


Figure 2.5: The five levels of the intentional structure

- **dialogue move** level. On this level, sequences of dialogue acts are combined into dialogue moves, like *greet* and *initiative*. Section 2.6.2 describes our definition and usages of moves.
- **dialogue game** level. Here, we combine the moves into games. Example of games are *greeting* and *negotiation*. Our definition and usage of the games are discussed in section 2.6.3.
- **dialogue phase** level. The games form the phases (see Section 2.6.4). As indicated in, e. g., (Kowtko et al., 1993), a dialogue can be described by *dialogue phases*. The phases are discussed in section 2.6.4.
- **dialogue** level. This level consists of just one part of the structure: the top level. In the case of VERBMOBIL (assuming cooperative negotiation dialogue) this level corresponds to, i. e., APPOINTMENT_SCHEDULING or PHASE_2_SCENARIO_COMPLETE (see section 2.6.4).

The first level above - the dialogue acts - has emerged partially by corpus analysis as well as literature studies. Along with the project, the corpus collection effort (see section 1.1) resulted in an impressive collection of material which was used for numerous tasks. One was the identification of typical communicative acts which we call dialogue acts. The dialogue acts have been defined in (Alexandersson et al., 1998). This document also served as manual for the annotation effort.

2.6.1 Dialogue Acts in VerbMobil

From both a theoretical as well as a practical point of view, the annotation of utterances or segments with labels, such as dialogue acts or communicative functions, is a well-established technique for analyzing communication and building dialogue systems. For dialogue modeling and one of the translation tracks in VerbMobil, the basic processing/translation entity is the dialogue act. As in many other schemata, our dialogue acts are task-oriented. Their purpose is to mirror the intention behind the dialogue phenomena occuring in our negotiation dialogues.

In general, language has to be *segmented* into *chunks* which can be processed according to the task of the system. For written (European) language one can use markers, like a full stop, comma or question marks to perform this.[17] The result is a sequence of sentences, and/or part(s) of sentences. Spontaneously spoken language contains nothing like full stops. Instead other means (i. e., prosodic analysis) for distinguishing where to segment the input have to be used. In the VerbMobil corpus, many contributions can be divided into sentences or phrase-like pieces of linguistic material. But our corpus shows frequent occurrences of non-grammatical material—in the sense of not being described as traditional grammar—which might still be pragmatically meaningful or even important for the dialogue (Allwood, 2001; Ward, 2000). The dialogue act has proved to be a helpful tool for assigning these chunks meaning. In VerbMobil it is used for different purposes, like translation (Reithinger, 1999; Reithinger & Engel, 2000), disambiguation for the semantic transfer (Buschbeck-Wolf, 1997; Emele et al., 2000) and for the generation of summaries (Alexandersson et al., 2000). In section 3.7 we discuss our usage of dialogue acts for discourse processing.

Figure 2.6 shows the hierarchy of dialogue acts used in the VerbMobil project (Alexandersson et al., 1998). The hierarchy is formed as a decision

[17] Japanese is one example for which this is not true. Instead other means, like sentence final particles have to be used.

tree for supporting the annotation of dialogues as well as processing. It divides the dialogue acts into three main classes:

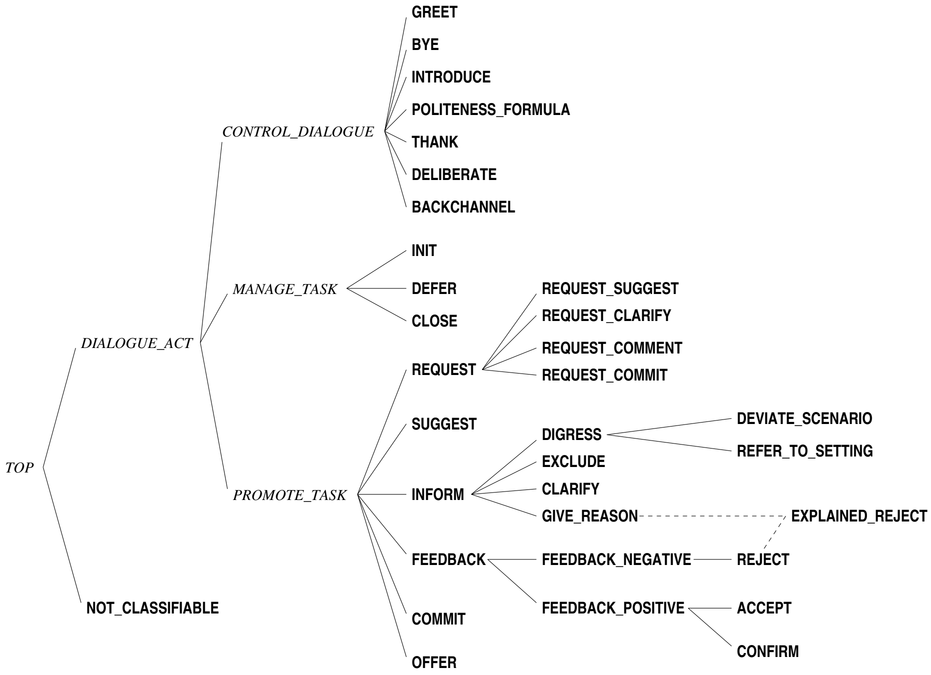


Figure 2.6: Dialogue acts hierarchy as employed in VerbMobil 2

- **control_dialogue** this set contains dialogue acts for segments concerned with social interaction or segments related to the dialogue itself (e. g. opening or closing the conversation) or segments used for smoothing the communication

- **manage_task** these acts are for segments concerned with managing the task (i. e. initializing, deferring, closing the task)

- **promote_task** finally, we use this set for utterances concerned with promoting the task.

A complete and very detailed description of the definition of the dialogue acts (including examples in German, English and Japanese) and how to utilize the hierarchy as a decision tree is found in (Alexandersson et al., 1998),[18] but see also (Alexandersson et al., 2000). In the appendix, an example of a dialogue act definition taken from (Alexandersson et al., 1998, page 77–78) where dialogue act SUGGEST is reprinted.

Discussion

The definition and usage of dialogue acts in VERBMOBIL has been inspired by a number of theories. Indeed, Austin and Searle were the pioneers (Austin, 1962; Searle, 1969) arguing that the intention of an utterance can be represented by labels. Bunt's definition of dialogue acts are very much inspired by the work of Allwood. A closer inspection of the relationship between our dialogue acts and those of Bunt (Bunt, 2000) shows that one of the three facets of his dialogue acts - the *linguistic form* - is outside our definition. Instead, Levinsons view has been adopted (Levinson, 1983). In (Levinson, 1983) it is argued that utterances can be solely described by something called *the illocutionary force* (together with propositional content) (see Section 2.3.1 on Page 27). Finally, our usage and definition of dialogue acts coincide with the first of Allwood's five facets of his *communicative acts* (Allwood, 1977), namely the intention and purpose of the utterance. The main reason for the relatively poor resemblance between our work and that of Allwood is that his view on communication is more general and takes more into account than that of ours. On the other hand, the frameword of Allwood is more tailored towards the study than the implementation of dialogue systems. For example, it is very unclear how a computer system equipped with speech and prosody recognition only, should be able to capture the "notion of conventional force".

However, the dialogue acts do not suffice when it comes to describing and processing our dialogues. The main challenge lies in the nature of spontaneous speech: many contributions contain repetitions, hesitations, thinking-aloud and the like. Still, the main force(s) of the contribution does not change. Also, a sequence of dialogue acts can contribute to form a single force or unit. Our solution are three additional levels of modeling: *dialogue moves*, *dialogue games* and *dialogue phases* as described below.

[18] (Alexandersson et al., 1998) does not contain the description of `inform_feature`. This dialogue act was introduced at the very end of the project in an effort to discriminate scenario relevant informs from "deviative" informs, i. e., informs outside the core scenario.

2.6.2 Dialogue Moves in VerbMobil

There are several facts and observations motivating the moves layer of the intentional structure. Among the more prominent we have: our data show frequent occurrences of sequences of utterances which together can be viewed as a unit. This unit is detected and used by (human) interpreters, in that they *reduce* the contribution into its central parts. Our next level in the intentional structure is concerned with this modeling - *dialogue moves*. Additionally we define the moves in such a way that, given the move, it should be possible to decipher who is having the initiative on the task level. We discuss the difference between our usage of moves and its usage in other contexts.

Consider the following example (translation in *italics*):

Example 1 (cdrom15par: g009ac)

ABA018:
initiative

INIT SUGGEST –
(4) ja , am nächsten Tag hm dann zurück
yes , on next day hm the back
yes return on the next day

SUGGEST -
(5) hm ja, da können wir eigentlich dann
hm yes, the could we really then
ja schon mittags fliegen
(PART) yet noon fly
hm well, then we could in fact fly at noon

SUGGEST -
(6) oder wir könnten natürlich auch noch
or we could of course also still
den Nachmittag nutzen, um in
the afternoon use, (PART) in
Hannover irgendwas zu unternehmen?
Hanover something to do?
or we could of course use the afternoon,
to do something in in Hanover

ABD019:
response

ACCEPT - (7) ja ,
yes ,

ACCEPT - (8) gerne ,
I'd love to

ACCEPT - (9) machen wir das
make we that
let us do that

INFORM - (10) doch . da gibt's doch Einiges
however . there are still some things
zum anschauen .
to look at
sure . there are quite a few things to look at

ABA020:
initiative

SUGGEST - (11) und dann könnten wir so gegen acht
and then could we around eight
oder gegen neun zurückfliegen.
or around nine fly back
how about flying back at about eight or nine

□

In turn **ABA018** the speaker takes the initiative not just by introducing the topic, but also by suggesting flying at noon. This is followed by another, alternative suggestion. These utterances form the first move – the initiative. The second speaker accepts this second suggestion with three utterances of type ACCEPT, and makes an additional remark that identifies which one of the suggestions was accepted.

Our data is full of another kind of examples which is yet another reason for introducing the moves layer. In these examples, the interlocutor uses different, sometimes even culture specific means of realizing her intention. Whereas the (american) English interlocutors tend to be very direct while rejecting a proposal more or less directly:

(12) MGT017: <uhm> <;comma> <B> <Smack> <B> that is not going to work . <B> <;seos> REJECT

it is in the Japanese culture impossible—unless one wants to be impolite—to reject a suggestion in such a way. Instead, there is a standard surface structure for reject a propsal which consists of several dialogue acts.

The following example is take from (Siegel, 1996, page 35) where a negotiation dialogue between a German and a Japanese is interpreted. The German interlocutor has suggested to meet on Monday Morning which turns out to be impossible for the Japanese dialogue partner. Since the Japanese culture forbids the American direct style of rejection, the interlocutor chooses the following surface structure:

(13) e̲to getsubôbi no gozenchû wa desu ne watashitachi no tokoro wa kore
Monday NO morning WA COP NE we WA this
kyûjitsu ni natte orimahite kaisha yasumi na mon de desu ne.dame nan
holiday NI get (Polite) company free COP NE bad
desu ne.sumimasen
COP NE excuse

which translates to: *Well Monday Morning* (FEEDBACK), *that is a holiday at our side* (GIVE_REASON), *the company is closed* (GIVE_REASON), *that is bad* (REJECT), *(I'm) sorry* (POLITENESS_FORMULA).

This sequence of dialogue acts[19] appearantly expresses a (Japanese) refusal. Still, the core intention of the contribution is the same as in 12. Hence to model our dialogues in a language independent way, we need to abstract away from the culture or language dependent realization. The concrete a pattern in 13 is absent in the (American) English part of our corpus. A very polite German interlocutor might utter something similar, but since the VERBMOBIL dialogues have a more colloquial style, we rather find direct rejects in the German part of the corpus too.

For a characterization of a *move* within a negotiation scenario we use the labels initiative, response, transfer-initiative and confirm. Central for the characterization is the concept of the forward or backward looking aspects of an utterance, e. g., (Allwood, 1994). By backward looking aspect we mean that – possibly a part of – the turn contains a direct reaction to something, which in the case of negotiation dialogues is, e. g., a proposal of a meeting place introduced earlier in the dialogue. The forward looking aspect roughly covers the cases where something was proposed or a new topic was introduced, which opens up a new discourse segment.

The main classes for the negotiation part of a dialogue are:

- initiative A (part of a) turn is annotated with the label initiative when

[19] In (Siegel, 1996), the sequence of dialogue acts is FEEDBACK GIVE_REASON GIVE_REASON REJECT_DATE APOLOGIZE. The dialogue acts used in (Siegel, 1996) are the set of dialogue acts used in the first phase of VERBMOBIL (see (Jekat et al., 1995)). The set presented in this thesis contains no REJECT_DATE but the more general REJECT. Jekat et al.'s APOLOGIZE is a special case of Alexandersson et al.'s POLITENESS_FORMULA.

the turn has a forward looking aspect and (i) when something is suggested and the dialogue contains no open topics, or (ii) when a suggestion refines a previous proposal that has been accepted explicitly, or (iii) when a direct counter proposal is made.

- response A (part of a) turn is annotated with the label response when the turn has a backward looking aspect. This occurs (i) when some earlier proposal is rejected or accepted, or (ii) when a declarative or imperative suggestion with an implicit acceptance or rejection contains a refinement of an earlier proposal.

- transfer-Initiative A (part of a) turn is annotated with the label transfer-Initiative when the turn has a forward looking aspect and (ii) when a suggestion is explicitly requested, or (i) when a topic is introduced without the locutor making a suggestion.

- confirm A (part of a) turn is annotated with the label confirm[20] when the turn has a backward looking aspect and (i) when a preceding acceptance is confirmed, or (ii) when a summarization of the agreement achieved so far is accepted.

We use the following moves for the beginning and the end of the dialogues:

- greet A (part of a) turn is annotated with the label greet if the speaker greets and optional introduces herself.

- bye A (part of a) turn is annotated with the label bye if the speaker saying goodbye.

Additionally, we introduce the following moves for the modeling of clarification dialogues:

- clarify-query A (part of a) turn is annotated with the label clarify-query for the cases where a clarification question is performed. This move can have a backward as well as a forward looking aspect.

- clarify-answer A (part of a) turn is annotated with the label clarify-answer for the case where an answer to a clarification question is given. This move can always be characterized by a backward looking aspect.

[20] Throughout this thesis, the distinction between the dialogue act CONFIRM and the move confirm is indicated by the type style.

We use some additional moves which are outside the pure negotiation but which serve additional phenomena occuring in our corpus:

- request-commit A (part of a) turn is annotated with the label request-commit for the cases where there is a request for commitment, i. e., taking care of booking hotel or table.
- commit A (part of a) turn is annotated with the label commit for the case when the speaker commits herself to take care of some action, e. g., booking hotel.
- request-describe A (part of a) turn is annotated with the label request-describe for the cases where a description of something is requested.
- describe A (part of a) turn is annotated with the label describe for the case where the speaker provide a description of something.

As for the dialogue acts, we give an example of the definition and instruction for the annotation of moves. In the appendix we give the definition of the move initiative.

Discussion

Although pointing in the right direction, we believe that the notion of initiative in (Linell & Gustavsson, 1987) is a bit diffuse. In their work a participant is taking the initiative by "adding something to the discourse." "Something" is too coarse for our purpose, and we allow ourselves to depart from such a general framework. Instead, our usage of initiative and response resembles one of the two levels - the task level - described in (Chu-Carroll & Brown, 1997), where the initiative is modeled on two levels. Chu-Carroll and Brown argued that it is possible, and indeed useful, to separate the task and the dialogue initiative. For the sake of the generation of minutes and summaries we are rather interested in how objects under negotiation are introduced and treated by the dialogue participants.

The work of Walker and Whittaker is more concerned with the relation between the control and attentional state within task-oriented and advice-giving dialogues. In (Walker & Whittaker, 1990) it is even questioned whether the notion of control is the same as having the initiative.

Our usage of forward- and backward-looking aspects of a move is related to the concept of *obligated* (backward-looking) and *obligating* (forward-looking) functions of a contribution, e. g., (Allwood, 1994). As presented in (Allwood, 1994), his *contribution* seems to resemble what we call a turn and is thus potentially a bigger chunk than our moves. The obligating and obligated function is more general than our forward/backward-looking aspect in that it has more facets. Whereas our aspects are focused on the attitude towards the task level, Allwood's functions are - as most parts of his frame work - multi-functional.

The unique feature of our framework is the (forward looking) move transfer-initiative - we argue that it is indeed possible to, on the task level, deliberately *hand over* the initiative as well as take it. This becomes evident when a direct request to suggest, for instance, a date is performed (see (15) in figure 2.7 on page 62). This feature of our model is absent in other approaches known to us. Rather, the initiative is said to be taken by one of the dialogue participants, but never unhanded. This might be due to the importance of taking the initiative for the sake of reaching some goal faster in a man–machine dialogue. According to our experiences with the VerbMobil corpus, this is not necessarily the case. Instead the speakers take their time, act politely, and sometimes even give the initiative away.

...

(14) wir wollen ja da zusammen nach Hannover fahren zu unserm Geschäftspartner
we want there together to Hanover going to our business partner
We should go to our business partner in Hanover

(15) und ich wollte Sie jetzt mal fragen wann es Ihnen denn prinzipiell
and I wanted You now *particle* ask when it You then in principal
am besten passen würde
best suit would
and I'd like to ask when it would suit you

(16) also mögliche Termine wären bei mir auf den ersten Blick einerseits
well possible dates would be to me on the first glance on the one side
zwischen dem vierten Juni und dem sechsten Juni oder aber zwischen dem
between the fourth June and the sixth Jund or but between the ...
Well, it would be possible between the fourth and sixth of June or between the ...

...

Figure 2.7: An example of unhanding. In (15), the speaker is requesting a suggestion. Such an act corresponds to the transfer-initiative move.

For the modeling of initiative, there are a number of subtle cases which are more difficult to grasp and define occurring, e. g., where a suggestion from one speaker is very general. The initiative is then, in a narrow sense and according to our definition, taken by the speaker. But the speaker, in this case, is not driving the negotiation that adamantly. However, this observation is non-trivial to quantify and to capture by a sharp definition. Thus, for the sake of simplicity and robustness during processing, we have chosen not to model different *levels* of initiative.

Finally, our choice of using "move" as the term for the modeling described above might be sub-standard since there is already a different concept or term in the literature called *move*. That move, however, resembles what we call a dialogue act and originates from the work of language games (Wittgenstein, 1974) and conversational games (Sinclair & Coulthard, 1975). Although we have searched for another term, we find that "move" best fits our intention and modeling.

2.6.3 Games in VerbMobil

With the moves as building blocks we can compose *games*. Games are described by context-free rules, where the left-hand side denotes the name of the game and the right-hand side consists of at least one move. The first is called the opening and the rest - one or more moves - are called the closing. For the course of cooperative negotiation dialogues we use the following basic games:

- *Introduction* This game corresponds to the INIT move. It consists of the turn(s) concerned with the topic introduction of the negotiation.
- *Negotiation* This game is the most characteristic game for our negotiation dialogues. A negotiation game always encompasses an initiative. However, the initiative might be preceded with a transfer-initiative, and succeeded with a response and, finally, one or more optional confirm(s).
- *Closing* This dialogue game, typically found at the end of the entire dialogue, signalizes that the speaker is concluding the conversation. Commonly precluding the actual farewell, this game may also be used to signalize the transition between the different dialogue topics such as accommodation and transportation.

Additionally the following phenomena occur in our corpus:

- *Opening* This game is used for the mutual greeting. It usually consists of greets, but might also contain clarifications.
- *Exit* This game signalizes the end of the conversation. The game *Exit* includes, e. g., the move Bye.

- *Nego-Incomplete* This game is reserved for cases within the dialogue where a certain negotiation between speakers has begun but for some reason, for example an inquiry or an abrupt topic change, has not been completed.

- *Exchange* This dialogue game is used to categorize the exchange of information between speakers that does not contain the kind of explicit suggestions and rejections typical of the *Negotiation* game but rather the small talk back and forth between speakers.

- *Commit* This game covers the case where the willingness to take care of certain aspects of our scenario, i. e., book actions is expressed.

In the appendix is the definition of the *Negotiation* game taken from the annotation manual.

Discussion

The concept of "opening move" or "initiating move" and "responding move" used for the modeling of English classroom discourse (Sinclair & Coulthard, 1975) or the map task (Carletta et al., 1997) has served as an inspiration to our model. Since our model differs a bit we avoid nomenclature conflicts by saying that the left part of the game is called *opening move* and the right, responding part is called *closing move*. This, since in our model, the number of closing moves is not restricted to exactly one move but is allowed to be zero, one or more. If we compare our work with that of (Carletta et al., 1997), we see that their moves correspond to our dialogue acts whereas their conversational games correspond to our games. Their concept of transactions is absent in our model. Furthermore their frame work is tightly coupled with the goal of the game - the move starting with an initiating move is then completed with its responding move when the goal of the initiating move has been fulfilled. As the authors admit, it is very difficult for a human annotator faced with the transcription of the whole dialogue to annotate such a structure reliably (see (Carletta et al., 1997, p. 10) for more details).

There is evidence for the usefulness as well as for negative influences from more complex structures when it comes to the recognition of, for example, dialogue acts. (Qu, Di Eugenio, Lavie, Levin, & Rose, 1997) describe an experiment for the inclution of higher level structures into the recognition of dialogue acts in the JANUS speech-to-speech translation system. There, the *cumulative error*, i. e., errors due to prior errors, seems to be more problematic if more context is taken into account for the recognition of dialogue acts. In (Poesio & Mikheev, 1998) another experiment based on the Map task corpus is described. They suggest that the inclusion of hierarchical context as found in the map task corpus can positively affect the recognition. However, their recognition is based on a perfectly recognized and thus construed context.

(Eckert & Strube, 2000) use a simplified structure inspired by the work of, i. e., (Carletta et al., 1997a, 1997b) where the basic entities are dialogue acts clustered into *initiation* and *acknowledgement*. These two entities are then combined to *synchronizing units*. We believe that their concept is too generic for the focus of our model. Still, our *exchange* game serves a similar function in our model.

Our work can be viewed as a sub-model of the full flavour of the structures in the map task corpus. However, the phenomenon of cumulative error as described above has made us restrict the modeling of games to the initiative and its *immediate* response(s), possibly with a prepended transfer-initiative. Part of the dialogue concerned with the exchange of information outside the scope of the pure negotiation is modelled by the more general exchange game. The only embedded games are the clarification games. Higher level structures are found in our phase level (see below). These render the complex higher level structures used in the map task corpus although they are simpler.

The exchange level of dialogue model of SUNDIAL (Bilange, 1991) shares much resemblance with our games level. Especially their negotiation patterns are more or less the same as our negotiation game. However, our optional initial transfer-initiative is missing in their patterns. The dialogue model of LINLIN is simpler than ours. This model, however, has been developed with typed dialogues in mind.

Finally, our games resemble the "interaction plans" in the plan recognition model of (Iida & Arita, 1992).

2.6.4 Dialogue Phases in VerbMobil

Whereas the topmost level describes different scenarios, such as, "VM-scenario-complete", the fourth level of our intentional structure ties the games together. In our scenario, cooperative negotiation dialogues consist of a greeting phase followed by a negotiation phase. A negotiation phase is sometimes preceded by an initiation phase where the topic of negotiation is presented/discussed[21] and succeeded by a recapitulation of the topic's result. Finally, the dialogue ends with a good-bye phase possibly preceded by a closing phase where the result of the negotiation as a whole is recapitulated.

In VerbMobil, the following dialogue phases are definened and used:

- opening includes the greeting until the themes has been introduced.
- init includes the part where the topic of a negotiation is presented.
- negotiation includes the negotiation.
- closing includes the conclusion of the negotiation.
- bye includes the saying of good-bye.

2.7 Propositional Content

To model propositional content we use a ontological description language - *Discourse Language Representation* (dlr) (Koch, 1998; Alexandersson et al., 2000) - inspired by description logic languages such as KL-ONE, e. g. (Brachman & Schmolze, 1985) and CLASSIC, e. g., (Borgida, Brachman, McGuinness, & Resnick, 1989). Just as in such languages there are concepts known as the *A-Box* which can be instantiated to represent abstract and physical real world object, called the *T-Box.* A concept has a name and a fixed number of roles which can be filled with other objects belonging to some concept. Concepts are connected by multiple inheritance where a more special object may introduce new roles. We call relations within one object *sister-relations* or *sister-roles* and the objects of sister-relations *sister objects* or *sisters.* A DLR expression is called direx (DIscourse Representation Expression).

The basis for the modeling are *objects* and *situations.* Situations are used to model activities of the speakers, such as *events* and *actions*, e. g., journeys, meetings and different kind of moves. Objects on the other hand are entities in the world - abstract and concrete - like time (see below), different type of locations and seats.

As an example, figure 2.9 shows the *textual representation* of a direx expression for the sentence *well it is about the business meeting in Hanover.* Figure 2.10 shows the graphical representation for the same expression. The

[21] In our scenario the speakers receive stringent instructions on how to behave. Therefore, there is no discussion or negotiation when it comes to the topic; the topics are introduced by the speaker and the hearer immediately accepts it.

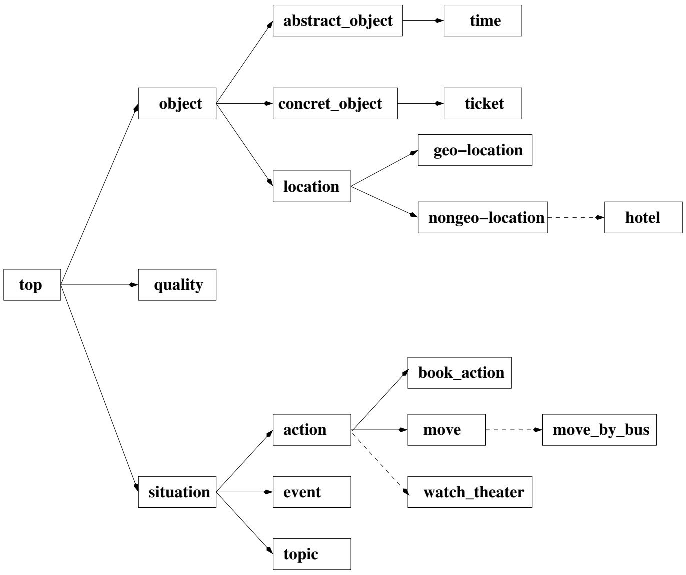


Figure 2.8: Clipping of the hierarchy

roles `has_det` and `has_loc_spec` are primarily used for translation purposes. These roles shows an example of sister-roles.

```
has_appointment:[appointment,
  has_meeting:[meeting,
    has_name='geschaeftstreffen'],
    has_location:[city,has_name='hannover',
    has_loc_spec=in,has_det=unknown]]
```

Figure 2.9: Textual representation of the propositional content of the sentence *well it is about the business meeting in Hanover*

Embedded in DLR is another language - *Temporal Expression Language* (TEL[22]) - which is used to model time (Endriss, 1998). TEL is a quite surface-oriented representation language but can be used as an interlingual representation.[23] and contains the necessary inferences (see below) in the domain of

[22] TEL is the successor of ZeitGram (Küssner & Stede, 1995).

[23] There are some expressions in German, like `morgens` which are ambiguous as well as not directly translatable into English. Its meaning is either `''in the morning''` or "`every morning`."

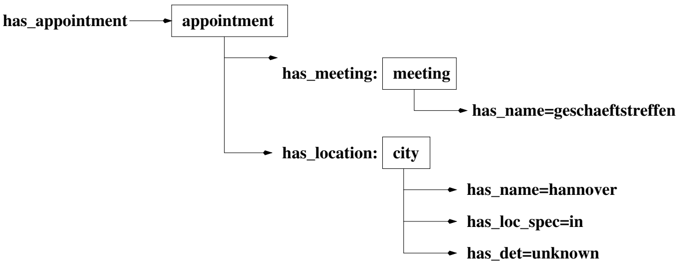


Figure 2.10: Graphical representation of the proposition content for the sentence *well it is about the business meeting in Hanover*

appointment scheduling. TEL is based on `TYPE:VALUE` pairs, e. g., *Monday at eight o'clock in the morning* is represented as an unordered list `[dow:mon, pod:morning, tod:08:00]`. TEL is based on a huge corpus analysis of English and German dialogues in the VERBMOBIL corpus and contains a wide range of constructs which are divided into five main constructions (see (Alexandersson et al., 2000)):

Simple comprises day, day of week, part of day, seasons, holidays etc.

Modified comprises qualifications, such as, *early* or fuzzy expressions such as *around two o'clock.*

Spans comprises durations or open spans

Referenced comprises time expressions with reference to some point, such as, *two hours from now.*

Counted comprises constructions for expressions like *the second week of June.*

Expressions in TEL are called TEMPEX (TEMPporal EXpression). Below are some examples of tempex expressions taken from our corpus[24]:

(17) *How 'bout the afternoon of Monday the ninth?*

`during:[pod:afternoon,dow:mon,dom:9]`

(18) *How 'bout the afternoon of Monday the ninth?*

`during:[pod:afternoon,dow:mon,dom:9]`

[24] See (Endriss, 1998) for more examples

(19) *I could do it on a weekend, from Friday the sixth of May to Saturday the seventh of May.*

```
interval:[pow:weekend,
  min_between([dow:fri,dom:6,month:may],
    [dow:sat,dom:7,month:may])]
```

(20) *What about twenty-ninth through to the second of April?"*

```
interval:min_between(dom:29,[dom:2,month:apr])
```

(21) *We need to get another meeting going for about two hours in the next few weeks.*

```
[for:fuzzy_dur(dur(2,hours)),
  during:next(int:dur(several, weeks))]
```

(22) *Are you free on Monday the twenty-sixth after twelve o'clock or how 'bout sometime in the morning on Tuesday the twenty-seventh?*

```
from:set([[dow:mon,dom:26,ex_after(tod:12:0)],
  [pod:morning,dow:tue,dom:27]])
```

For the representation of larger chunks of discourse and for disambiguation we divide utterances as belonging to one of four *topics*:

Scheduling for the actual date schedule (begin and end time, location etc).

Traveling for the travel arrangements (departure information, means of transportation etc).

Accommodation for hosting (rooms, prices etc).

Entertainment for spare time activities (visiting a cinema, restaurant etc).

Now we have the tools to represent utterances: The dialogue act, topic, DLR and TEL. Here are some more examples taken from our corpus together with their textual representation:

(23) `das ist ja am zwanzigsten Januar um elf Uhr vormittags`
that is on the twentieth of january at eleven o'clock in the morning

```
[SUGGEST,
          uncertain_scheduling,
          has_date:[date,tempex='tempex(ge_2920_0,
          [from:[dom:20,month:jan,tod:11:0,pod:morning_ger2]])']]
```

(24) *so we have to leave Munich at six o'clock*

```
[SUGGEST,
        scheduling,
        has_move:[move,has_source_location:
        [city,has_name='muenchen'],has_departure_time:
        [date,tempex='tempex(en_2920_0,[from:tod:6:0])']]]
```

(25) *so how about a three o'clock flight out of Hannover*

```
[SUGGEST,
        traveling,
        [has_move:[move,has_departure_time:{time_of_day:3},
        has_source_location:[city,has_name='hannover']]]
```

Discussion

We have strived to make our modeling of propositional content language independent and usable for discourse processing and translation. The KL-ONE like description language is expressive enough to model the content of not just utterances but bigger chunks of discourse by abstracting away from stylistic and linguistic features in favour of a goal-oriented representation. Since the concepts of our ontology are products of an extensive corpus analysis we have a model which is deeply anchored in the domain. While this is an advantage for discourse processing, a drawback of such a comparatively coarse-grained model is, of course, that we lose, e. g., linguistic information while instantiating the domain objects. However, we can represent the result of a complete dialogue with our language (see chapter 3 and 4). Finally, it is, using typed frame-based representations, possible to add information to the discourse state in a formal way using default unification like operations (see section 3.7.3 and 3.7.4). This is very important, since our version of default unification—the OVERLAY operation—has turned out to be the fundamental operation for discourse processing.

Our approach to modeling propositional content shares some similarities with that of the C-Star consortium (Levin, Gates, Lavie, & Waibel, 1998). There, an interlingua-approach for a translation scenario is presented where an expression in the so-called *interchange format* is called *domain action* (DA). In Addition to concepts and arguments, a DA contains a speech act. A DA has the following syntax:

$$\textit{speaker} : \textit{speech act} + \textit{concept}* \ \textit{argument}* \tag{2.4}$$

The speaker tag is either "`a:`" for agent, or "`c:`" for customer. The speech act is a single label.[25] Concepts and arguments can appear in arbitrary number one after another. A total number of 38 speech acts, 68 concepts, and 86 arguments have been defined yielding 423 DAs. An example of the representation is:

(26) On the twelfth we have a single and a double available
```
a:give-information+availability+room
(room-type=(single & double), time=(md12))
```

Admittedly, there are certain linguistic phenomena not covered by their approach, i. e., relative clauses, modality, politeness, anaphora, and number. An evaluation, however, yielded a coverage of 92% measured on two previous unseen dialogues.

Clearly, this approach is simpler than ours. It is also unclear if their approach is expressive enough to represent the result of a complete negotiation of a topic, or if the representation is just aimed at representing utterances. Also, their language lacks constructions for representing anaphora.

2.8 Conclusion

We have presented and motivated the design of our modeling of the negotiation dialogues for the VerbMobil scenario. The modeling is based on three pillars:

- **The dialogue act**

  The dialogue act is our primary tool for characterizing utterances in VerbMobil. The dialogue act is used to capture the illocutionary force of an utterance. We described how the design of the dialogue act has been influenced by previous literature works, such as those of Austin, Searle, Levinson, Allwood and Bunt.

- **The propositional content**

  Using widely accepted as well as new ideas, the second pillar for representing utterances and discourse is the propositional content (Alexandersson et al., 2000; Koch, 1999). Our representation language - DLR - was inspired by knowledge representation languages, like KL-ONE (Brachman & Schmolze, 1985). Embedded in DLR there is a language - TEL - for representing time expression. The goal of the design—this

[25] There are three exceptions: The speech acts “verify”, “request-verification”, and “negate” combine with other speech acts. “*So you’re not leaving on Friday, right?* has the speech act “request-verification-negate-give-information”.

goes for the dialogue acts as well—has been to represent information encoded in the surface structure of language in such an abstract way that we have, in principal, a representation which can be used as an interlingua for translation and additional tasks, such as, resolving elipses. The most prominent are translation, analysis and reasoning as well as generation.

- **The intentional structure**

  Based on the utterances annotated with dialogue acts, we have designed a tree structure that we call the *intentional structure*. The requirements from the transfer module (i. e., supporting the translation process by distinguishing the dialogue phase) as well as robustness when it comes to processing has a big impact on the design of the structure. However, previous work in the area of mixed initiative, initiative-response, e. g., (Linell & Gustavsson, 1987; Chu-Carroll & Brown, 1997; Walker & Whittaker, 1990; Kowtko et al., 1993) has also been indispensable. Most notably, we have a characterized a phenomenon in a, to our knowledge, new way: the *moves layer* of the intentional structure. There, we view parts of contributions as chunks expressing the same *move type*. However, the notion of move in this thesis differs from the traditional use. To us, the move represents a sequence of utterances which together form a certain communicative abstract function, e. g., initiative and response. The traditional usage of the term move is what we call the dialogue act. One of the moves – Transfer-Initiative – is new and not found anywhere else in the literature. The Transfer-Initiative move resembles what is known as *how may I help you?* utterances, where the system starts by unhanding the initiative.

In the next chapter we will present the algorithms and tools used in the running system.

Chapter 3

Dialogue Management in VerbMobil

This chapter is devoted to the implementation of the dialogue component of VerbMobil which we will refer to as DiVe. We start by presenting the tasks of the module and the difference between the tasks our module are faced with in comparison to those of dialogue managers of man–machine dialogue systems. Selected parts of both mono-lingual as well as multilingual dialogues show the variability in the VerbMobil data.

3.1 Introduction

Apart from, or maybe in addition to, imperfect input, at least two other factors influence our approach to the understanding and processing of VerbMobil dialogues:

- users of the running system have to deal with inaccurate translations and therefore abundantly use repetitions, confirmations, and clarifications. The ideal conditions during the recording of, for instance, the mono-lingual dialogues from our corpus, drastically change the users behaviour. Our mono-lingual part of the corpus does not show ample examples of clarification dialogues due to translation problems. This trait is easy to explain, but does not change the fact that we lack data covering phenomena like clarifications caused by, e. g., one participant suspects a false translation.
- VerbMobil mediates the dialogue and, as indicated below (see section 3.4), is not supposed to perform too frequent clarification dia-

logues. It has to obey the principle of *unobtrusiveness*. As a result, we pursue an approach using methods not unlike those of Information Extraction (i. e. (Hobbs et al., 1996)): we know what to expect, we try to extract as much information as possible, checking consistency on the way.

To make it more clear what the consequences are for processing within a mediating scenario, the interpretation of (partial) utterances in a man–machine system is heavily guided by, e. g., the expectations of the system. Many systems[1] employ a rather simple way of managing the dialogue. For instance, a well known technique for getting the user to provide information needed for, e. g., a data base look-up is for the system to take the initiative and ask precise questions.

There are, of course, other differences between the VerbMobil dialogues and man–machine dialogues. Amongst the more striking ones, the size and complexity of a user contribution in the VerbMobil corpus is on average larger.

3.2 Characteristics of the VerbMobil Dialogues

This section surveys the characteristics of the VerbMobil data from the point of view of dialogue management. As will be shown, the experiences gained during recording of wizard-of-oz dialogues, led us take a mediating approach to dialogue management. Other important characteristics are the variability in verbosity and complexity of user contributions.

3.2.1 Human–human vs. man–machine negotiation dialogue

In contrast to man–machine dialogues, our corpus contains many dialogues where the turns does not consist of one or two utterances, but up until ten or even more. The average number of utterances in our corpus is 2.2 for all dialogues (English and German) annotated with dialogue acts and the distribution of the number of utterances or dialogue acts per turn is shown in table 3.1. The distribution for English, German and Japanese monolingual dialogues are shown in the appendix (pages 195–197). The average turn length is 2.4 for the German dialogues, 1.9 for the English, and 2.5 for the japanese ones. Contrary to the German and English dialogues,

[1] There are, of course, systems employing real mixed initiative, i. e., the user might at any point say anything.

the Japanese dialogues has a lower number of turns with one dialogue act than turns with two dialogue acts. This shows that the structure of turns is more complex in particular for the Japanese–Japanese dialogues but also for the German–German dialogues than the English–English (see the appendix for more details).

10 sample multi-lingual (German–English) dialogues have been annotated with dialogue acts. The distribution is shown in figure 3 in the appendix. For these dialogues, the average turn length is 1.8. The number of multi-lingual dialogues are, however, to small to draw any conclusions. Still, the curves indicates a more complex structure than the monolignual dialogues in that the number of turns with 2 dialogue acts are twice as many as the one with one dialogue act.

3.2.2 Length of the dialogues

Table 3.1 shows the annotated CD-ROMs—"*# CD-ROMs*", the number of annotated dialogues—"*# dial.*" and dialogue acts—"*# dial.acts*", and the mean, minimal, and maximal length of dialogues, measured in dialogue acts.

Table 3.1: Annotated CD-ROMs

language	*# CD-ROMs*	# dial.	*# dial. acts*	*mean*	*min*	*max*
German	13	738	37954	51.24	6	208
English	6	375	22682	59.93	7	347
Japanese	2	402	15574	39.52	16	83
Sum	21	1505	76210	50.41	6	347

3.2.3 Behavioural differences due to cultural differences

- While German speakers tend to be polite and offers the dialogue partner to take the initiative and pose a suggestion, the American English speakers are very direct and efficient, often presenting parts of their calendar asking the partner to select one of the proposed dates.

- While our English parts of the corpus contain rather short turns, the German contain a lot of phenomena like thinking aloud and thereby producing, e. g., suggestions which are immediately rejected by the same speaker in the same turn.

- Apart from the big difference between Japanese and European languages, the way one behaves within the different cultures is different too in that Japanese speakers are very polite. For negotiations, the way one negotiates is (for us Europeans) somewhat extreme: there are no direct rejections since it is impolite to reject a suggestion outright. Instead, the following pattern is common: the suggestion is repeated (feedback) followed by at least one explanation why it is impossible to accept the suggestion. Not until this point is the actual rejection uttered and is followed by an obligatory excuse (Siegel, 1996, page 35). This negotiation pattern is possible in the German culture but rare[2] and absent in the (American) English speaking culture.

3.2.4 Turn complexity

Some turns are rather complex as shown by the utterances 27 – 36. This turn also shows another particularity in our data. The speaker tends to think out loud, verbalizing utterances which humans can filter due to, e. g., prosodic characteristics and lack of (new or interesting) information, but, which are hard for a machine to interpret as deliberations or small talk.

(27) ja also
well

(28) Zu- nach Hannover ist die Verbindung von M"unchen aus auch an und f"ur sich auch recht g"unstig,
Trai- to Hanover is the connection from Munich also by itself also pretty good

(29) deshalb , im $I-$C-$E ist ’s auch an und f"ur sich kein Problem.
therefore , in ICE is it no problem

(30) w"ar’ mir auch sehr recht.
that would be okay for me

(31) das heißt , <"ah> ab f"unf Uhr morgens etwa fahren Z"uge
that is , eh the trains will go starting at 5am

(32) viereinhalb Stunden sind die unterwegs ,
the duration is 4.5 hours ,

(33) das heißt , wir k"onnten am fr"uhen Mittag da sein , selbst , wenn wir nicht allzu fr"uh hier losfahren.
that is we could be there before noon , even though we do not leave to early

(34) f"unf , ich rechne grade , f"unf , um zehn .
five , I’m counting , five , at ten

(35) sollten wir versuchen , da wir gegen zw"olf dort sind,
should we try , to be there at twelve

(36) dann k"onnten wir ja um sechs Uhr sechsundf"unfzig hier wegfahren.
then we could leave at 6:56

[2] In fact, we have not found a single occurence in our corpus. This finding is, however, not a proof.

3.2.5 Subdialogues

Utterance 37 – 47 below shows an excerpt from a dialogue with a relatively simple structure containing a clarification dialogue. These kinds of clarifications are more or less absent in our mono-lingual corpus. For this excerpt, the challenge for a system tracking the dialogue is to understand that the Hotels introduced in the first utterance are later, after the clarification dialogue, picked up and finally agreed upon as the `Hotel Luisenhof`, and not a dropped negotiation topic.

...

(37) `ja ich kenne mehrere Hotels in Hannover`
I know several hotels in Hanover

(38) `good`
`can you make reservations for us there ?`

(39) `können Sie das bitte wiederholen ?`
could you please repeat that ?

(40) `yes`
`will you please make reservations at a hotel ?`

(41) `ja`
yes

(42) `welches Hotel von den dreien möchten wir nehmen ?`
which one of the three hotels should we take ?

(43) you choose .

(44) `ich schlage das Hotel Luisenhof vor`
I suggest the hotel Luisenhof

(45) `good`

(46) `I trust your choice`

(47) `why don't we meet at the station on Wednesday ?`

...

3.2.6 Controlling vs. Mediating

As a result of the mediating scenario, our module cannot serve as a "dialogue controller" like the dialogue manager in man–machine dialogue systems. The only exception is when clarification dialogues are necessary between VerbMobil and a user. However, too frequent clarification dialogues turned out to be so inconvenient that the users tended to avoid using VerbMobil. The excerpt below is taken from one of the WOZ dialogues. There, the German speaker (ZIE) is trying to arrange a dinner with his English colocutor (DAP). In this case the wizard was instructed to frequently perform system-initiated clarification dialogues.

(48) ZIE142: `danach könnten wir gemeinsam Abendessen gehen`
Afterwards we can have dinner .

(49) VERBMOBIL: `Bitte wiederholen Sie Ihre "Au"serung`
Please repeat that
`Es ist ein Fehler in der semantischen Verarbeitung aufgetreten`
An error occurred in the semantic processing .

(50) ZIE144: `danach k"onnten wir gemeinsam abendessen` x *Afterwards we could have dinner together*

(51) VM: `Bitte wiederholen Sie Ihre "Au"serung mit anderen W"ortern`
Please repeat that with other wordings .
`Es ist ein Fehler in der semantischen Verarbeitung aufgetreten`
The semantic processing was not successful

(52) DAP147: `maybe a bit louder ?`

(53) ZIE144: `yes , I invite you for the dinner`

3.2.7 Conclusions

With these experiences as background, the project decided not to initiate clarification dialogues too frequently. Consequently the dialogue module (henceforth DM) took the approach of tracking the dialogue: instead of intervening, the course of the dialogue has to be obeyed no matter how strange it might look. For man–machine systems, the system is allowed to take the initiative and clarify strange and unexpected information. We are, however, not allowed to do that. The system as a whole, should just translate on demand, and not intervene. Such requirements put extra demands for robustness and flexibility on the components of a discourse processor.

3.3 Recognizing Spontaneous Speech

We have seen how the VERBMOBIL data actually looks. Now we turn to the processing of spontaneously spoken language. The recognition of spontaneously spoken language in VERBMOBIL was based on two types of recognizers: speech recognition for the recognition of words, and prosody recognition which was used to detect, e. g., sentence and segment boundaries and emphasized words.

It is a well known fact that today's speech recognizers are imperfect. The effect of imperfect recognition is not, as a naive reader might think, just that speech recognition might result in a sequence of words corresponding to erroneously syntactic constructions. A common phenomenon is syntactic correct construction with erroneously recognized words. To cope with the absence of delimiters as found in and used for processing of written text, prosody recognition has been used. Also this component uses a statistic model trained on our labelled corpus. The most prominent negative effect of non-perfect prosody recognition is false segmentation.

3.3.1 Speech Recognition

Table 3.2 (Wahlster, 2000) shows how speech recognition emerged during the VerbMobil project. As it started in 1993, state-of-the-art speech recognition technology allowed for speaker dependent isolated-word recognition using push-to-talk technology. For continuous recognition, a relatively small lexicon was used. At the end of the project, we deployed speaker independent, speaker adaptive, large vocabulary, continuous speech recognition for spontaneous speech.

Table 3.2: Development of Speech Recognition during the VerbMobil project

	Input Conditions	**Naturalness**	**Adaptability**	**Dialogue Capabilities**
1990	Close Speaking Microphone/Headset, Push-to-talk	Isolated Words	Speaker Dependent	Monologue Dictation
↓	Telephone, Pause-based Segmentation	Read Continuous Speech	Speaker Independent	Information seeking Dialogue
2000	Open Microphone, GSM Quality	Spontaneous Speech	Speaker Adaptive	Multi-party Negotiation
	VerbMobil			

Modern techniques like speaker adaption allows for a reduction in the error rate within the range of 5% for most speakers, yielding a word error rate of 3 – 20% (average 7.2%) (Young, 1996). But even these techniques do not change the fact that today's speech recognition does not provide perfect recognition. The final VerbMobil evaluation reveals an average word accuracy for about 75% for a lexicon with 10000 entries (Malenke, Bäumler, & Paulus, 2000). This is comparable with other results, e. g., the JANUS project reports on a word accuracy of 65% for Spanish (Gates et al., 1996). The vocabulary size is not mentioned. Word recognition rates on telephone conversations in the Switchboard corpus are around 50%. (Kemp, Weber, & Waibel, 2000) report on a word error rate of below 25% for recognizing the German daily news on TV (Tagesschau) within an information retrieval scenario. However, for their task this rather poor performance seems to be sufficient.

Nevertheless, for the interpretation and understanding of the spoken language even small errors in speech recognition can have fatal consequences. The following samples are taken from our evaluation corpus (the first row contains transcriptions - the second the recognized words):

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

(55) When would be a good time for us to meet?
One would be a good time for us to meet.

(56) Good so we will leave Hamburg on the first
I would so we were to leave Hamburg on the first

Utterance 54 will be interpreted as a rejection of December as a possible date in contrast to the spoken suggestion (since having a meeting usually signals an explained rejection of a date). Utterance 55 triggers the false suggestion of one o'clock for the meeting instead of a request for a suggestion. Even though the initial part of the last utterance 56 is corrupted, the content can be recovered. Utterances 54 and 55 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 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 example 56 is interpreted correctly by our shallow analysis approach (see section 3.5).

In a broader perspective, a recognition mistake as the one above is a member of a broader type of processing errors we have coined *confabulations*. Confabulations are processing errors resulting in made-up entities which "appear", for instance in the summary. We show in chapter 4 that speech recognition is not the only place where the system hallucinates discourse objects.

3.3.2 Prosody

In spoken language there is no punctuation like exclamation marks or full stops.[3] Moreover, for the running system the instruction for manual segmentation of our corpus does not apply either. For instance, we decided not to count the number of finite verbs and use this information as an additional hint for segmentation. Again, this is due to the fact that speech recognition is not 100%. Instead we use the output of the prosody module (Batliner, Buckow, Niemann, Nöth, & Warnke, 2000). This module distinguishes between several boundaries, such as B0 – "normal word boundary", B2 – "intermediate phrase boundary", B9 "Non-grammatical" boundary, e. g., hesitation or repair, and Q3 – "question mood".

For our purposes, the recognition of the so-called **D3** boundaries are most interesting since they, when recognized properly, chunk the continuous signal into *segments* suitable for the annotation of dialogue acts. A segment eventually coincides with the speech signal between dialogue act boundaries. Actually, the prosody module trained the D3 model on the same corpus as the dialogue act recognizer (see section 3.5.1). An evaluation (Batliner et al., 2000) shows that the D3 boundaries can be recognized with a recall of 84–89%—depending on whether part-of-speech (POS) information has been used or not—for German, 97% for English and 81% Japanese.

The following excerpt (57) shows the use of, for example the usage of the *B9* and the *D3* boundaries.

(57) ...at all `B3 M3 D3 <A>` and in the `<L> B9 <P>` thirty fourth week `B3 M3 <P> <A>` the would...

To indicate what consequences false segmentation can have, we show an example taken from our corpus where 58a and 58b show the recognized machine segmentation and 59a and 59b the human segmentation based on the transcription of the spoken utterance.

(58) a. `repeat please do you have time`
b. `on the fifth and sixth why don't you.`

[3] To our knowledge, the only person who actually used to—he has been dead since 23.12.00—speak the delimiters aloud is Victor Borge, but since he never used the system—he is also a Dane–we have no hope for such indications for the running system.

(59) a. *OK, could you repeat please.*

b. *Do you have time on the fifth and sixth, or don't you?*

Here, the challenge is to interpret segment 58a as a request to "repeat the time", and the second as a suggestion. For this particular example, it is actually possible to extract the relevant information, i. e., dialogue act and propositional content, using the methods described below in section 3.5.1 and 3.5.2.

3.3.3 Discussion

The examples above show on the one hand how fatal even one single erroneously recognized word can be. On the other hand, more heavily distorted output from the recognizer is still possible to process. We conclude that modules involved with interpretation must use robust methods like those in (Hobbs et al., 1996). Also, the dialogue module has to provide flexible and robust processing techniques. It is not possible to trust the input too much. Such an approach may have the consequence that the recognition of future events is based on false assumptions, especially for systems which have no way to perform system-initiated clarifications.

In fact, in the VerbMobil setting some of the recognition errors survive until the dialogue is finished. They even appear in the summaries and in chapter 4 we will show how frequent this phenomenon is.

In (Kemp et al., 2000), an evaluation of a system using speech recognition for recognizing the German daily news (Tagesschau) is presented. There, despite a word error rate below 25% the task of indexing and retrieving functions surprisingly well. Similar observations were made during the 10.000 turn evaluation of VerbMobil (Tessiore & Hahn, 2000). Although the word recognition rate dropped considerably, the translation worked relatively well. Both these results might indicate that, for the system as a whole, perfect recognition is not compulsory. Therefore other guidelines for evaluation are proposed, e. g., (Gates et al., 1996; Levin et al., 2000), namely how well, in this case, the translation system works when it comes to solving the task. Instead of focusing the evaluation on speech or dialogue act recognition, the success of the overall task is evaluated.

3.4 Architecture and Tasks of the Dialogue Component

We now turn to the description of the actual implementation of the dialogue module. First, we describe the tasks and the environment of the module within the VerbMobil system. A short recapitulation of the input structures precedes the description on how to recognize dialogue acts (section 3.5.1) and propositional content (section 3.5.2).

Due to its role as information server in the overall VerbMobil system, we started collecting requirements from other components in the system early in the project. The result divides into three subtasks:

- we allow for other components to *store* and *retrieve* context information.
- we draw *inferences* on the basis of our input.
- we *predict* what is going to happen next.

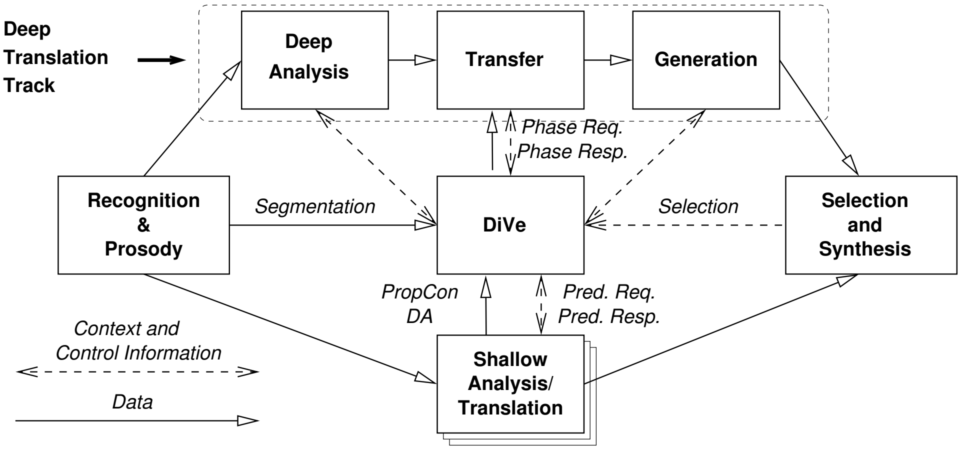


Figure 3.1: The dialogue component and its neighbour modules in VerbMobil

Within VerbMobil there are different processing tracks: parallel to the deep, linguistic based processing, different shallow processing modules also enter information into, and retrieve it from the dialogue module. The data from these parallel tracks must be consistently stored and made accessible in a uniform manner (see Figure 3.1). In this thesis, however, we will concentrate on the following core technologies developed during the VerbMobil project:

top down predictions As proved in, e. g., (Reithinger, 1995; Reithinger, Engel, Kipp, & Klesen, 1996), the analysis of dialogue acts are enhanced by using top down predictions.

disambiguation For the translation task, we developed methods for organizing context in such a way that the transfer component (Emele et al., 2000) could be helped with the disambiguation of meaning or readings of expressions or words (Alexandersson et al., 1998).

minutes and summaries One of the goals of our efforts in the second phase of VerbMobil was to produce summaries (see chapter 4) and minutes of the dialogue (Alexandersson & Poller, 1998, 2000; Reithinger, Kipp, Engel, & Alexandersson, 2000). The idea behind the summaries is to provide the user with a document stating what had been agreed on, thus recapitulating the *result* of the negotiation. The minutes serves as an abstracted recapitulation of the *progress* of the dialogue. In this document, the central content of each turn is recapitulated, i. e., hesitations and repetitions were removed.

In the final VerbMobil system, one additional functionality was implemented where the *wordings* of the negotiation was compiled into a similar document as the minutes. Since this task consists of merely collecting and formatting the processed input chains or selected output wordings respectively (depending on target language) we will not expand on this topic.

3.5 Input to the dialogue component

In chapter 2 we described our representation and modeling of user contributions: direx and dialogue acts. In the running system, one of the translation tracks called *syndialog* (Reithinger & Engel, 2000) provides us with this input. The technique for recognizing dialogue acts is borrowed from the field of language modeling. Since this approach is used in other processing steps within the dialogue module, we present this more thoroughly (section 3.5.1). The recognition of propositional content (section 3.5.2) is given a more brief presentation.

3.5.1 Recognition of the Dialogue Act

For the recognition of the dialogue act, n-gram models are used. They are robust, and as far as our experience goes, more accurate and more reliable than hand-crafted rule-based recognition. The strength of this technique is clear when the output of the speech recognizer drops.

N-gram models

In the dialogue module of VerbMobil, n-gram models (Jelinek, 1990) has been used for several purposes:

- **prediction**

  In order to *predict* the next dialogue act to come, deleted interpolation of n-gram frequencies was used, where the interpolation weights are determined by a slightly modified version of the original algorithm (Reithinger et al., 1996). In order to *compute* the most probable dialogue act, d_j, to come, the following formula is used:

  $$P(d_j|\, d_1, \ldots, d_{j-2}, d_{j-1}) = \max_{d}\; P(d|\, d_1, \ldots, d_{j-2}, d_{j-1})$$

  where $d_1, \ldots, d_{j-1}$ are the dialogue acts of the preceding utterances. An evaluation yielded a prediction rate of about $40\%, 65\%$, and 75% if the actual dialogue act was within the first, second, and third most probable predictions.

- **classification**

  To recognize, e. g., dialogue acts for an utterance, a statistical method taking a sequence of words as input was used (Reithinger & Klesen, 1997). The formula used for computing the dialogue act is linear interpolation of uni- and bi-grams:

  $$D = \operatorname*{argmax}_{D'}\; P(W|\, D')\; P(D'|\, H)$$

  where W is the string of words, D is the dialogue act, and H is the dialogue act history. An evaluation showed that the recall and precision rate for negotiation acts, e. g., SUGGEST, ACCEPT, and REJECT, lies between 60% and 80% for both German and English. Dialogue acts composed of more or less fixed phrases, like GREET, and THANK, were very well recognized, i. e., 80%–100% recall and prediction, whereas dialogue acts that describe deviations from the actual negotiation, like

Table 3.3: Distribution of dialogue acts and length information for the German dialogues.

	German				
dialogue act	*# (%)*	mul	*min*	*max*	*mwl*
ACCEPT	8643 (23)	4.39	1	52	4.401
BYE	1621 (4)	2.84	1	23	5.838
CLOSE	399 (1)	7.49	2	33	4.451
COMMIT	249 (1)	8.67	1	37	4.623
DEFER	143 (0)	10.55	3	24	4.571
GIVE_REASON	1438 (4)	9.68	1	49	4.748
GREET	1407 (4)	3.63	1	15	4.456
INFORM	6002 (16)	7.52	1	52	4.763
INFORM_FEATURE	843 (2)	10.44	1	41	5.195
INIT	1702 (4)	12.43	1	37	5.139
INTRODUCE	709 (2)	5.78	1	22	4.166
POLITENESS_FORMULA	342 (1)	5.79	1	19	4.418
REJECT	3084 (8)	7.89	1	55	4.604
REQUEST	870 (2)	7.01	1	25	4.742
REQUEST_COMMENT	1177 (3)	5.62	1	33	4.143
REQUEST_COMMIT	62 (0)	8.15	3	21	4.881
REQUEST_SUGGEST	946 (2)	7.84	1	26	4.577
SUGGEST	7876 (21)	10.63	1	50	5.019
THANK	441 (1)	3.11	1	10	4.702

CLARIFY and MOTIVATE, were poorly recognized. Finally, exploiting the dialogue act history improved the recognition rate by 3% (Reithinger & Klesen, 1997).

In the running system, the set of dialogue acts as shown in 2.6 has been reduced to 19 acts (Reithinger, 2000). There are two main reasons for this:

- 10 dialogue acts together cover less than 1% of the annotated utterances, and contain acts like DEVIATE_SCENARIO. They neither contribute to the negotiation dialogues nor do they control the dialogue like the DEFER. DEFER is a rare dialogue act but is essential for the dialogues.

- The other 6 acts show a hight degree of confusion with other, closely related acts, e. g., FEEDBACK_POSITIVE with ACCEPT. For these two acts, the main distinction is that, contrary to the latter, the former does not carry propositional content.

We developed tools to create and analyse confusion matrices amongst others to detect such cases.

Table 3.4: Distribution of dialogue acts and length information for the English dialogues.

	English				
dialogue act	*# (%)*	mul	*min*	*max*	*mwl*
ACCEPT	6557 (29)	2.83	1	55	3.890
BYE	625 (3)	3.59	1	19	3.270
CLOSE	70 (0)	6.71	1	15	3.562
COMMIT	158 (1)	10.27	1	31	3.948
DEFER	74 (0)	13.49	4	36	3.688
GIVE_REASON	491 (2)	11.47	3	49	3.794
GREET	276 (1)	1.89	1	7	4.122
INFORM	4446 (20)	7.16	1	54	3.716
INFORM_FEATURE	1831 (8)	11.45	1	58	3.989
INIT	701 (3)	11.90	2	42	3.910
INTRODUCE	35 (0)	4.11	1	7	3.722
POLITENESS_FORMULA	246 (1)	4.19	1	27	3.435
REJECT	1311 (6)	8.58	1	33	3.737
REQUEST	986 (4)	6.86	1	35	3.905
REQUEST_COMMENT	551 (2)	5.15	1	30	3.650
REQUEST_COMMIT	31 (0)	10.10	1	17	4.086
REQUEST_SUGGEST	452 (2)	8.71	1	33	3.784
SUGGEST	3735 (16)	10.32	1	42	4.011
THANK	106 (0)	2.49	1	16	4.284

Table 3.5: Distribution of dialogue acts and length information for the Japanese dialogues.

	Japanese				
dialogue act	*# (%)*	mul	*min*	*max*	*mwl*
ACCEPT	3272 (21)	5.25	1	41	3.893
BYE	476 (3)	2.62	1	11	7.561
CLOSE	n.a. (n.a.)	n.a.	n.a.	n.a.	n.a.
COMMIT	n.a. (n.a.)	n.a.	n.a.	n.a.	n.a.
DEFER	n.a. (n.a.)	n.a.	n.a.	n.a.	n.a.
GIVE_REASON	817 (5)	13.39	3	39	3.997
GREET	197 (1)	3.27	1	10	5.533
INFORM	2618 (17)	9.50	1	44	3.906
INFORM_FEATURE	n.a. (n.a.)	n.a.	n.a.	n.a.	n.a.
INIT	577 (4)	18.79	2	64	3.886
INTRODUCE	1181 (8)	4.87	2	21	5.215
POLITENESS_FORMULA	1180 (8)	5.72	1	19	4.734
REJECT	667 (4)	9.54	1	37	4.190
REQUEST	906 (6)	9.56	2	43	4.081
REQUEST_COMMENT	632 (4)	6.87	2	77	4.140
REQUEST_COMMIT	n.a. (n.a.)	n.a.	n.a.	n.a.	n.a.
REQUEST_SUGGEST	663 (4)	10.02	3	31	4.015
SUGGEST	2251 (14)	14.09	3	47	3.934
THANK	137 (1)	4.19	2	11	5.692

Table 3.6: Distribution of dialogue acts and length information for all data.

	All Data				
dialogue act	*# (%)*	mul	*min*	*max*	*mwl*
ACCEPT	18472 (24)	3.99	1	55	4.154
BYE	2722 (4)	2.97	1	23	5.391
CLOSE	469 (1)	7.37	1	33	4.330
COMMIT	407 (1)	9.29	1	37	4.333
DEFER	217 (0)	11.55	3	36	4.219
GIVE_REASON	2746 (4)	11.10	1	49	4.302
GREET	1880 (2)	3.34	1	15	4.539
INFORM	13066 (17)	7.79	1	54	4.227
INFORM_FEATURE	2674 (4)	11.13	1	58	4.346
INIT	2980 (4)	13.54	1	64	4.548
INTRODUCE	1925 (3)	5.19	1	22	4.764
POLITENESS_FORMULA	1768 (2)	5.52	1	27	4.533
REJECT	5062 (7)	8.28	1	55	4.309
REQUEST	2762 (4)	7.79	1	43	4.213
REQUEST_COMMENT	2360 (3)	5.84	1	77	4.041
REQUEST_COMMIT	93 (0)	8.80	1	21	4.577
REQUEST_SUGGEST	2061 (3)	8.73	1	33	4.196
SUGGEST	13862 (18)	11.11	1	50	4.543
THANK	684 (1)	3.23	1	16	4.909

Table 3.3 – 3.6 contain the details about the dialogue acts: the total number and percentage *#(%)*, the mean utterance length in words *mul*, the minimal *min* and maximal *max* length in words, and the mean word length in characters *mwl*. The Japanese data were annotated using the second revision of our dialogue act annotation scheme. This did not include the dialogue acts close,commit, defer, inform_feature, and request_commit. The Roman transcription is based on syntactical and morphological criteria. Phrases ("bunsetsu") are separated by spaces and are considered as words for further processing.

In the appendix, the recognition result for real dialogues are shown, e. g., our sample dialogue.

3.5.2 Recognition of Propositional Content

As indicated above, the syndialog module uses a "dialogue-act based translation" approach for translation.[4] This module exploits another technology borrowed from yet another field of research. Motivated by its speed, robustness and ease with which it is managed, finite state transducers (FST)

[4] The work described in this section is presented for the reason of completeness. For more reading on this topic, the reader is referred to (Reithinger & Engel, 2000).

(Hobbs et al., 1996) was used for the recognition of propositional content (Reithinger & Engel, 2000). This approach is one of the reasons for a translation quality comparable to competing translation tracks within the system. No evaluation of recognition rate has been performed. For longer sentences though, the authors conclude:

> "A drawback of this approach is that complex sentences, e. g., if-then-sentences, are often analyzed incorrectly."

However, the advantage of using FSTs has been proven by its ability to analyze distorted utterances as shown in example 60 below. Consider the acoustic best chain

(60) I would so we were to leave Hamburg on the first

where the *good so we will* was recognized incorrectly as *I would so we were.* A DIREX representation of the above utterance is:

```
[INFORM,has_move:[move,
  has_source_location:[city,has_name='hamburg'],
  has_departure_time:[date,time='day:1']]]
```

More examples of the performance of this component are found in the appendix.

3.6 Dialogue Processing in VerbMobil - DiVe

Dialogue processing in VerbMobil consists of managing three structures. The three structures are:

- **The Sequence Memory** consists of records mirroring the user contribution as processed in the running system. For each user contribution—turn—a turn record is added and for each segment as segmented by the prosody module (see section 3.3.2) a segment record is added. Each turn and segment receives a unique identifier by the underlying test bed. Using these labels we can access the corresponding turn and segment. Furthermore, we define a successor and predecessor relation on both turns and segments, allowing for the traversal of the structure in both directions (see Figure 3.8).

- **The Thematic Structure** is used to represent the propositional state of the negotiation by relating new DIREX-es to the prior context. At the end of the dialogue, this structure contains all proposals and objects—accepted and rejected—as introduced in the negotiation. These are represented as instances of our domain model.

- **The Intentional Structure** is used to infer the dialogue phase by building a tree structure as described in section 2.6. The leaves of the structure connects to the segment records of the sequence memory, and the root represents the type of negotiation dialogue, e. g., "complete negotiation dialogue" or "negotiation dialogue without greet or bye".

Two distinct engines are responsible for the management of the two latter structures. The *dialogue processor* manages the thematic structure and the *plan processor* manages the intentional structure. Both engines make use of the sequence memory which is used for storing turn-relevant and segment-relevant information, both temporary and persistent. Figure 3.2 depicts the main components – both structural and functional – of the dialogue module.

For maintaining the above-mentioned structures, new input from the syndialog module causes the dialogue module to process the following steps:

1. If it is the first segment of a new turn, then add a new turn to the sequence memory.
2. Add a new segment to the sequence memory and store the dialogue acts and DIREX in the segment. Possibly re-interpret the dialogue act.
3. Update the thematic structure.
4. Update the intentional structure.

Below, the functioning of the two above-mentioned engines is described: section 3.7 describes step 3 and step 4 is described in section 3.8.

3.7 Managing the Thematic Structure

The thematic structure[5] is used to track the negotiated objects during the dialogue. The usefulness of this work will be exemplified in chapter

[5] The work presented in this section is joint work with Michael Kipp, Tilman Becker and Norbert Pfleger. I'm very grateful to them for allowing me to use the results in this thesis. Much of the material in this section has been presented elsewhere, e. g., (Kipp, Alexandersson, & Reithinger, 1999; Kipp, Alexandersson, Engel, & Reithinger, 2000; Reithinger et al., 2000), (Alexandersson & Becker, 2001, 2003b) and (Pfleger, Alexandersson, & Becker, 2002; Pfleger, 2002)

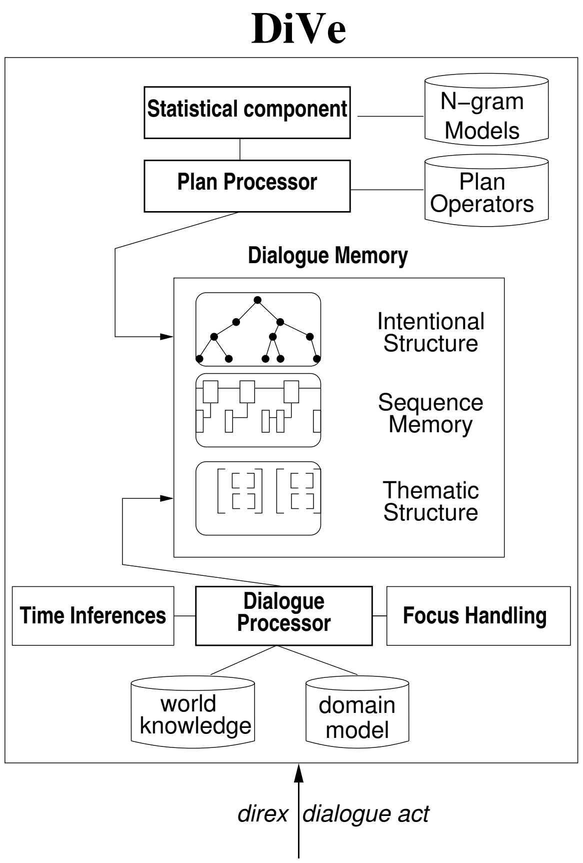


Figure 3.2: Architecture of DiVe (Dialogue Processing for VerbMobil)

4. The basic ideas about discourse modeling developed for the dialogue module have been taken over for the discourse modeling module in the SmartKom project. However, some of the facets—especially the completeness algorithm—has been further developed and, most notably, formalized. We present the refined version of this algorithm—the overlay algorithm—in section 3.7.4.

An important mechanism for almost every dialogue system is the ability to maintain context in such a way, that additional information is added to, or (partly) overwrites old information in a correct way. The main feature of the thematic structure is to maintain such dialogue information. This is carried out on the basis of our domain model. From the syndialog module, the dialogue module receives an abstract representation of the utterances as presented in section 2.7. Goal of the processing is—like for most computer systems in this field of research—to be able to answer questions like *What is currently negotiated?* For the VERBMOBIL system, we have to answer an additional question: *What has been accepted/rejected?* The reason is the setting of the VERBMOBIL system; we are modeling negotiation dialogues. A positive side-effect of the tracking of the acceptance status of the negotiated objects is that we can use this information to generate summaries at the end of the dialogue (see chapter 4). It would actually be possible to generate summaries in the middle of the dialogue. This thread has not been investigated though.

During the the first phase of VERBMOBIL, we concentrated solely on time expressions and inferences to find out the speakers' attitudes towards those expressions (ACCEPT, REJECT, UPTAKE) (Maier, 1996; Alexandersson, Reithinger, & Maier, 1997). In the second phase of VERBMOBIL, the domain was extended to more complex suggestions, e. g., flights, hotel reservations, spare-time activities in the evening, e. g., (Kipp et al., 2000). Still, the basic assumptions remain: what possible actions the speakers could perform and what possible items could come up. We therefore introduce two notions:

- *negotiation objects* A negotiation object is a structure for keeping track of instances of the VERBMOBIL domain model, speaker attitudes and a possibly empty set of relations to other negotiation objects.

- *negotiation acts* A negotiation act is a grouping of different dialogue acts depending on what attitude they impose towards the negotiation objects.

Below we will explain these concepts in more detail but first we have to explain the notion of *topics.*

3.7.1 Topics

Topics partition our domain into four sub-domains:

- SCHEDULING The scheduling topic comprises meetings. There are two main types of meetings in our dialogues. The most obvious one is the actual reason for the negotiation, but the negotiation might also result in a meeting of a train station or a hotel.

- TRAVELING The traveling topic comprises the parts of the dialogue where the locutors negotiate the trip to and from the meeting. This sometimes includes a drive between the airport and the hotel. The traveling topic is the most complex one of our scenario.

- ACCOMMODATION The accommodation topic comprises the lodging part of the negotiation.
- ENTERTAINMENT The entertainment topic comprises the spare time activities. This includes, e. g., dinner or going to a movie.

For each topic we keep *templates* for all summary-relevant items, e. g. `journey` and `booking` for the topic TRAVELING. Templates are used to transform the speakers' content to structurally uniform *negotiation objects* that can be stored and compared for further summary processing (section 3.7.3).

Focus handling

Most processing is local to each topic. For focus handling, a simple and robust approach is used: each topic keeps its own list of negotiation objects and a focus list. The focus list keeps track of the most recently mentioned negotiation objects that are annotated accepted/rejected in case of feedback utterances (section 3.7.2).

Topic shifts are recognized by using rules that work on current topics, key-words, dialogue act and extracted content. They are managed by an algorithm we call "update_topic":

Definition 1 (update_topic)

```
   if   (this is the first utterance)               (3.1)
        then (take SCHEDULING as new topic)         (3.2)
elsif   (dialogue act is INIT)                      (3.3)
        then (determine new topic)                  (3.4)
elsif   (other topic than current one)              (3.5)
        then (check evidence)                       (3.6)
 else   (retain current topic)                      (3.7)
```

□

The test in 3.5 and 3.6 consists of testing key-words and the corresponding DIREX-es for indication of other topics. This knowledge has been collected through manual corpus analysis. The same is the case for 3.4, where the wordings and DIREX of the utterance indicate what topic to choose.

In the case of a topic shift, the respective focus list 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:

(61) A: so that was Monday the twenty-first at the check-in counter (SCHEDULING)

(62) B: I'll do the flight reservations (TRAVELING)

(63) A: and I will let my secretary take care of the hotel (ACCOMMODATION)

The respective items (flights, hotel) can be found on the local focus list of the respective topic frame. The topic shift in B's utterance is recognized using key-word spotting.

3.7.2 The Dialogue Processor

The dialogue processor works on the current topic frame and changes its state according to the users' actions. We have found, examining our corpus of sample data, that a negotiation essentially consists of four basic *negotiation acts* (see (Sidner, 1994) for similar distinctions) relevant for the management of the negotiation objects:

- *propose* - The dialogue acts SUGGEST, INIT, OFFER and COMMIT are those acts that introduce (new) objects in the negotiation. *Propose* acts roughly correspond to the initiative move.

- *feedback* - The dialogue acts ACCEPT, REJECT EXPLAINED_REJECT are those acts that contains a positive or negative attitude for an object of negotiation. *Feedback* roughly corresponds to the response move.

- *elaborate* - The dialogue act INFORM is often used as a means of extending the focused object. The object is implicitly accepted by the speaker, and an elaboration causes the speaker attitude to be set to accept for the elaborated object. There is no direct correspondence between *elaborate* and a move.

- *request* - The dialogue act REQUEST is often used to point at certain roles of the negotiated object (see 67 where the speaker points at the has_departure_time role) or sometimes to request a suggestion. The former case belongs to the initiative move and the latter case corresponds roughly to the Transfer-Initiative move.

These actions trigger a number of operations as shown in Table 3.7. As far as our experience goes, such an approach is superior to, e. g., finite state models in that they are more flexible. The reason is that it is almost impossible to predict what is going to happen next in human–human dialogue—even in such a restricted setting as in the VerbMobil project. Early in the project, we actually defined a finite state model for, e. g., the prediction of the next dialogue act to come. As it turned out, there are better and more flexible methods for such tasks, e. g., (Reithinger et al., 1996). We will return to the disadvantage with hand-crafted models in section 3.8.2.

Table 3.7: Mapping from dialog to negotiation act and operations

Dialog act	Negotiation act	Processing
SUGGEST, INIT OFFER, COMMIT	*propose*	(1) complete object (2) focus object
ACCEPT, REJECT EXPLAINED_REJECT	*feedback*	annotate focused object with acceptance/rejection
INFORM	*elaborate*	merge object with focused object
REQUEST	*request*	store object in temporary memory

Now, the main operations the dialogue manager needs to perform are

- **Complete utterance content:**

  Anaphoric references, ellipses and answers to direct requests have to be resolved. This is implicitly done by *converting* the DIREX to a negotiation object and then completing the resulting object with previous objects. This operation is discussed in section 3.7.3 and 3.7.4.

- **Annotate attitudes:**

  FEEDBACK acts by speakers give away their attitude (acceptance/rejection) towards the focused negotiation object. A *strong* accept/reject is an utterance that mentions the accepted/rejected proposal explicitly, e.g.:

  (64) `A: let’s meet on Tuesday then.`

  (65) `B: Tuesday is fine.`

  Confirmations are annotated as strong accepts, e.g.:

  (66) `A: so I’ll see you Tuesday, 2 o’clock in your office`

- **Collect negotiation objects:**

  For each topic all attitude-annotated negotiation objects are stored for further processing.

Question-answer pairs (REQUEST - INFORM) 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 the case of a positive reply, the propositional content is treated as if introduced as a fact at the time 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.

  (67) `A: when's that flight going?`
  → [plane, has_date: ?DATE]

  (68) `B: 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.

All operations described result in a list of attitude-annotated negotiation objects. This can be used as the basis for selecting the summary items (chapter 4). It is most important that each negotiation object be, in itself, as rich in information as possible in order for the summary to be complete. How this is achieved is described in the next section (3.7.3).

3.7.3 Completing the Data

Ellipses and anaphora are commonplace in everyday conversation. Detecting them is one problem, representing them is another. As for the representation, there are two principle approaches, one using links (which replace missing data) and inferring the complete object at a later stage, and the other using instant completion of the data. For an anaphora that means replacing the anaphora by the actual reference object and for an ellipsis that means adding the missing object(s) to the representation of the elliptical object.

Our approach for completion of data is based on the structure of our representation. Our (typed) frame-based representation makes it possible to adapt techniques similar to those put forward in, e. g., (Kaplan, 1987; Grover, Brew, Manandhar, & Moens, 1994). However, since the representation of time expressions is structurally different, the (structural) assumptions underlying our algorithm for the completion operation cannot be applied in a direct way, and has therefore been necessary to deploy special treatment for time expressions.

Time expressions

For the treatment of time expressions, a taxonomy of *temporal units* has been developed (Birkenhauer, 1998). First, however, the time expression has to be completed. A time expression is completed by using the path of instant completion as explained below. Whenever a time expression is encountered, the system tries to find a sponsoring expression and completes the new expression. The sponsoring expression is first looked up in, e. g., the focus or in the *situative context*, i. e., the time of the dialogue.

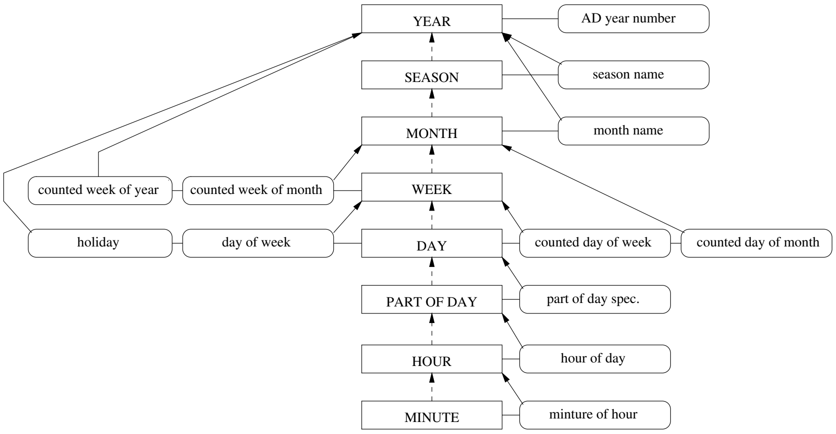


Figure 3.3: Principal temporal units (capital letters) and their possible specifications

Our approach is based on the temporal specification tree shown in figure 3.3. First of all, we define completeness of a time expression as a contiguous path from the most specific node (that would be *counted day of month* in example 61, page 94) to the root node (`year`). The completion of an incomplete time expression T using a potential sponsoring expression S from the focus list is conducted by copying those parts of S into T that are needed to construct a contiguous path from T's most specific node to its root node (see (Kipp et al., 2000) for a more formal account).

In our examples below we will use a simplified notation for the DIREX and TEMPEX formalism. TEMPEXes will appear in curly brackets. Those parts of an expression that have been taken over from a sponsor are underlined.

(69) **A:** Why not meet on the fourth of June?
⟶ {year:2000, month:june, day_of_month:4}

(70) **B:** The sixth would be better, I'm afraid.
⟶ {year:2000, month:june, day_of_month:6}

Here, A's time expression was completed with the year at the time of speaking (*situative context*), and B's expression was in turn completed by copying the month and year of A's expression.

(71) **A:** So is it going to be the eight or the ninth?
⟶ or({day_of_month:8}, {day_of_month:9})

(72) **B:** A Friday? Yes, Friday's good.
⟶ {day_of_month:9, day_of_week:friday}

71 and 72 show an expample of *coordination.* To correctly select the right alternative, the calendar component has to be consulted.

(73) **A:** Six o'clock looks like a good time for me.
⟶ {. . . , day_of_month:8, time_of_day:6:0})

(74) **B:** Couldn't we do it before?
⟶ {. . . , day_of_month:8, before({time_of_day:6:0})}

Modifiers, like before, after, around etc. are considered to be orthogonal to the domain of time expressions as modeled in figure 3.3. The critical issue in completion is how to choose the scope of the modifier.

Completion of DIREX expressions

We employ a similar approach with the content representations in the DIREX formalism with one additional initial step: If necessary, the input is converted to a negotiation object (C) using a set of templates. In the example in figure 3.4 (see page 100), the time expression is converted to a journey object. These templates play the role of plan based predictions in traditional dialogue systems. In (Löeckelt, Becker, Pfleger, & Alexandersson, 2002), we show how this conversion—called bridging—is performed in the SmartKom system. There, the predictions from the planner and the focus structure in combination with the information about who is having the initiative guides this operation. The overall goal—in VerbMobil as well as in SmartKom—is to obtain fully qualified negotiation objects (VerbMobil) or application objects (SmartKom) for completion. For a system like SmartKom, where the system is one of the dialogue partners and thus can seize the initiative, more precise information can be utilized.

In a next step, for the new object C we (1) find a suitable sponsor and (2) take over parts of the sponsor (see figure 3.4). Both steps are modeled by a single function `complete`(C,C_f) which tries to complete C using C_f as a sponsor, returning a boolean value for success or failure. By applying this function on every C_f on the focus list until it succeeds[6] we find a sponsor and complete C.

The function, `complete`, works recursively through the C object (and respective sub-objects of C_f). It first checks certain preconditions: named entities (cities, persons etc.) can only be sponsored by objects with equal name, move objects must have certain temporal properties (move back *after* move there) and so on. If the preconditions hold, all subtrees of C_f that do not occur in C are added to C (see figure 3.4) Under certain conditions relations can be specialized (e.g. `has_time` to `has_departure_time`). Note that since C_f is already a completed object, we obtain a complete object C without further processing of other preceding objects. Important for the work here is that we keep a specificity relation between the objects which makes it possible to retrieve the most specific objects.

3.7.4 Completing the Data - revisited

This section contains a formalization of the completion algorithm which we have given the name OVERLAY. Our formalization has received its name since it is convenient to think about the functioning of completion as putting or laying a structure representing new information—covering—over another (representing old information)—background—thereby possibly overwriting conflicting parts of the old information and inheriting old coherent information. An outstanding feature discriminating our approach from others is the usage of a formal scoring function. This is necessary since, contrary to the completion algorithm described above which either succeeds or fails, OVERLAY *always* succeeds and it is necessary to rank the different results.

[6] We found it useful to introduce an upper bound for the number of objects being tested by e.g. 3 (*recency threshold*).

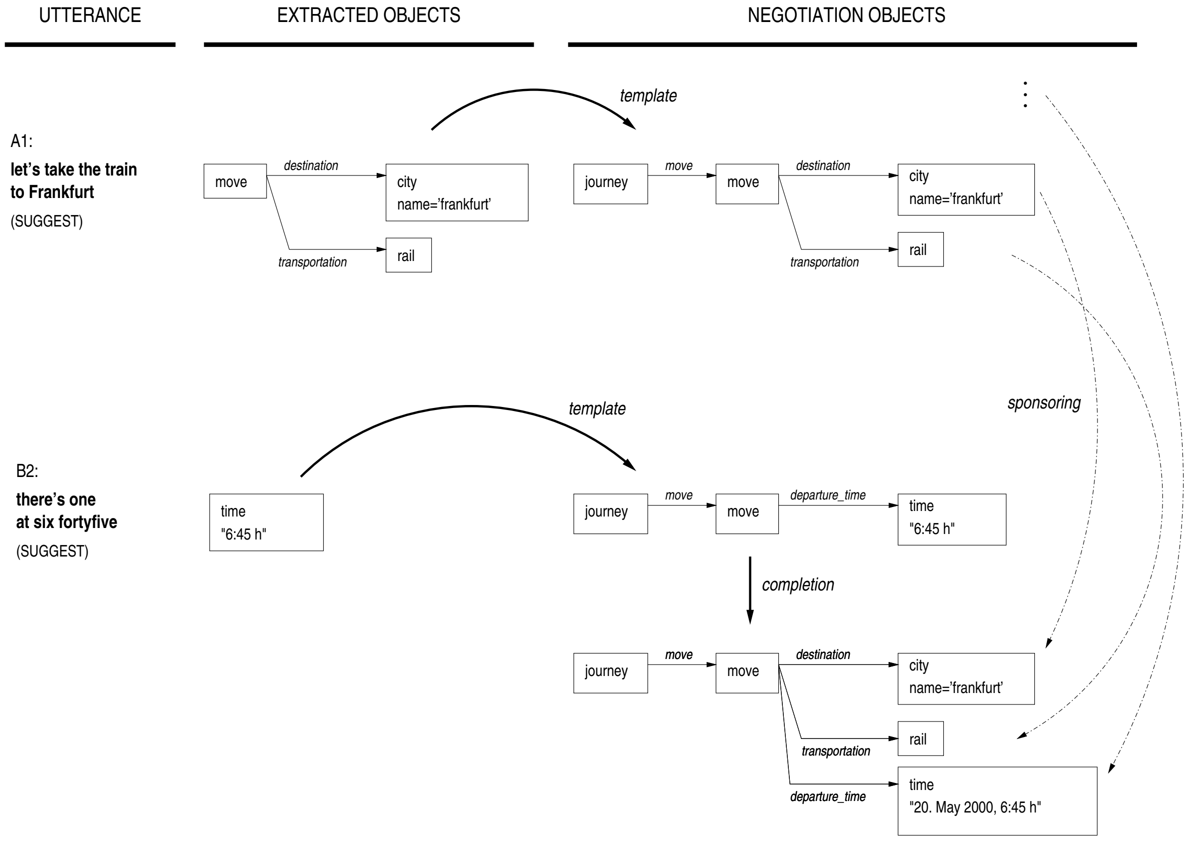


Figure 3.4: Dialogue excerpt showing recognized utterance (left), extracted objects (middle) and content objects (right) derived by template filling and completion with a sponsoring expression. The resulting structure is more specific than the arguments of the *complete* operation.

In the final VerbMobil system, the domain model – evolved over several years – consisted of two languages, namely direx and tel where the latter is part of the former. Furthermore, the modeling was inhomogeneous, which is apparent, e. g., by the usage of topic containers as described above. A more complete modeling of the domain would include the topic in the domain model. Therefore, it would—with the exception of the representation of time—be better to use *one* language for the modeling the domain.

The formalization described below is the product of the advancement of the technology developed in VerbMobil which has been successfully used in SmartKom, e. g., (Wahlster, Reithinger, & Blocher, 2001; Reithinger et al., 2003). In what follows we simplify the expressiveness of the representation language for encoding the domain. Instead of using description logics we will use, a somewhat restricted version of, typed feature structures (henceforth TFS) as described in, e. g., (Carpenter, 1992; Krieger, 1995). When it comes to expressiveness of the underlying representation language, there are other approaches advocating similar ideas, e. g., (Denecke, 1999).

Within this class of representation languages, we find some more or less related efforts coined under terms like *priority union* and *default unification.* In the succeeding two sections, we give an short summarization of these approaches (section 3.7.5) and relate our own work to these approaches and give a precise formulation (section (3.7.6).

3.7.5 Related Work

Default reasoning has received a vast amount of attention in the past (see, for instance, (Antoniou, 1999) for a survey of default logics and (Touretzky, 1986) for a comprehensive formal account for the mathematics of inheritance systems). The general reason for default reasoning is the need for making plausible conjectures given incomplete information. Such situations is common in our everyday life. Take, for instance, a emergency room where the doctor has to treat a patient possibly without knowing the real reason for the status of the patient; the doctor has to assume that certain facts are true without really knowing. Of course, it would be possible to await the result of blood tests etc but not doing something might cause the patient to die. Parallel examples are the question whether Tweety can fly given the information that Tweety is a bird. However, given the additionally information that Tweety is a penguin, we conjecture that Tweety cannot fly.

Within these theories, there is the distinction between credulous and skeptical inferences. Inferences the former type means that as many consistent conjectures as possible are drawn. Inferences of the latter type means that inferences are not drawn in case conflicts are at hand.

Within the computation linguistics community, operations similar to OVERLAY working on typed feature structures are known under terms like *priority union*, *lenient composition* and *default unification*. They all operate on a *strict* structure (Carpenter, 1993; Grover et al., 1994; Ninomiya, Miyao, & Tsujii, 2002) and a *default* or *defeasible* structure (Bouma, 1990; Carpenter, 1993; Ninomiya et al., 2002; Grover et al., 1994). The usage is broad and includes areas like default representation in the lexicon (Lascarides & Copestake, 1999), robust parsing (Ninomiya et al., 2002; Fouvry, 2003)) and discourse inference (Grover et al., 1994).

To our knowledge, the first mentioning and implementation of such an algorithm was part of the DPATR workbench and was called *clobber* (Karttunen, 1986, 1998). Clobber is similar to what is coined *priority union* in (Kaplan, 1987). In his effort of formalizing optimality theory, Karttunen names a similar operation *lenient composition*. The idea is the same: information in a default structure is added to a strict structure as long as the information does not conflict. Kaplan suggests using his priority union operator for either morphological operations where one value overwrites a default value or the resolution of elliptical constructions. The latter is picked up in (Grover et al., 1994) which elaborates on the work of (Prüst, 1992) (see below) for computing ellipses.

Kaplan does not tell us how to process coreferences nor typed feature structures. The former is tackled by the approache of, e. g., (Bouma, 1990). Bouma, e. g., (Bouma, 1990), gives a recursive definition of default unification. His idea is to

> “remove all default information which might lead to a unification conflict.”

Another definition similar, but slightly different in style and function is given by Carpenter. We concentrate on (Carpenter, 1993) where *credulous* and *skeptical* default unification are defined. The idea behind the credulous operation is to, given a lattice of TFSs inflated by subsumption, generalize the default structure until it unifies with the strict structure. Generalization can then be thought of as walking upwards[7] until a structure is found that meets this requirement. Credulous default unification is ambiguous since, in general, there might be several paths in the lattice leading to possibly different structures that unifies with the strict one.

[7] Remember that Carpenter views the lattice in the opposite way—a more general feature structure is found below.

Carpenter does not give an algorithm but a nice and generic formal characterization of his credulous default unification:

Definition 2 (Credulous Default Unification)

Let F and G be two TFS. Then, *credulous default unification*—$\overset{<}{\sqcup}_c$—is defined as

$$F \overset{<}{\sqcup}_c G = \{F \sqcup G' | G' \sqsubseteq G \textit{ is maximal such that } F \sqcup G' \textit{ is defined}\}$$

□

Carpenter's second definition—*skeptical* default unification—is based on the desire to obtain a unique result. The idea can be summarized as

> "... [maintaining] only default information which is not in any way conflicted."

This is achieved by computing the least upper bound[8] of the result of the credulous default unification:

Definition 3 (Skeptical Default Unification)

Let F and G be two TFS. Then, *skeptical default unification*—$\overset{<}{\sqcup}_s$—is defined as

$$F \overset{<}{\sqcup}_s G = \sqcap(F \overset{<}{\sqcup}_c G)$$

□

Besides the more general approaches of default unification as described above, there are approaches intended for special processing, e. g., parsing of ill-formed input. One of the earlier ones is (Imaichi & Matsumoto, 1995) where an extention to standard unification, a variant of *forced unification* called *cost-based unification* ($\overset{\rightarrow}{\sqcup}$) is introduced (Ninomiya et al., 2002). Their idea is to continue processing at unification failure, but punish the result by adding a cost in case of inconsistency. In the following example the symbol $\top$ represents inconsistency, and $\top\{sing, plur\}$ an inconsistent set:

$$\begin{bmatrix} \text{NUM:} & \textbf{sing} \\ \text{PERS:} & \textbf{third} \end{bmatrix} \overset{\rightarrow}{\sqcup} \begin{bmatrix} \text{NUM:} & \textbf{plur} \\ \text{PERS:} & \textbf{third} \end{bmatrix} = \begin{bmatrix} \text{NUM:} & \top\{\textbf{sing,plur}\} \\ \text{PERS:} & \textbf{third} \end{bmatrix}$$

Imaichi and Matsumoto mention several ways of defining costs, such as, the number of inconsistent sets.

[8] In Carpenters world greatest lower bound.

(Ninomiya et al., 2002) extend the work of (Imaichi & Matsumoto, 1995) and introduce the theoretically elegant *ideal lenient default unification* (ILDU). A pragmatic and efficient algorithm called "*lenient default unification* (LDU) is also provided. Its time complexity is linear to the size of the strict and the default feature structure. In contrast to Carpenter's credulous default unification, the goal of their algorithm is to maximize the information content of the *resulting* structure. Carpenter's credulous default unification tries to maximize the amount of information in the *default* feature structure.

Formally, the ideal lenient default unification is defined as

$$F \overset{>}{\sqcup}_{LDU} G = \sqcap \left\{ F \sqcup G' \;\middle|\; \begin{array}{c} G' \sqsubseteq_f (F \sqcup_f G) \textit{ is maximal} \\ \textit{such that } F \sqcup G' \textit{ is defined} \\ \textit{without the top type} \end{array} \right\}$$

where $\sqsubseteq_f$ is a subsumption relation where the top type is defined. The optimal answer for ILDU is computed by $F \overset{>}{\sqcup}_s (F \sqcup_f G)$ which has exponential time complexity (Copestake, 1993). As a realistic and fast alternative Ninomiya et al. introduces *lenient default unification* which is almost like ILDU but is based on two basic ideas:

1. inconsistency caused by path value specifications are replaced by generalizing the types at the fail points.
2. inconsistency caused by path equivalence specifications can be removed by unfolding the structure-sharing of fail path nodes.

Interestingly, Ninomiya et al. do not mention anything about the effect of introducing types into the feature structures. Instead they admit that the result of "*default unification*" does not necessarily produce a totally well typed structure indicating that types are of second interest.

Finally, another example of using default unification for robust parsing is given in (Fouvry, 2000, 2003). In particular and contrary to most of other approaches, the approach of Fouvry does not decide on which reading should be selected in case there are multiple ones.

3.7.6 Formalizing OVERLAY

Having worked with and worked out the basics of OVERLAY, we are convinced that its desired functioning when it comes to structural manipulations is the same or very similar to credulous default unification (Carpenter, 1993) as described above. Our experiences sofar are restricted to a domain model exploiting unary inheritance only. Also, our experiments and validation have involved structures without reentrance. Therefore, our formal definition will be based on a restricted subset of the typed feature structures as defined in (Carpenter, 1992; Krieger, 1995). This subset, however, is attractive since it

- is expressive enough for encoding a domain model for a large multimodal dialogue system for multiple domains - SmartKom

- it is possible to provide an efficient algorithm for overlay.

While the definition of credulous default unification would do as characterization of overlay, its definition provides no clue concerning its operational semantics. The literature provides some explanations for, at least, similar operations. However, the algorithm suggested in (Grover et al., 1994) is admittedly slow whereas the algorithm presented in (Ninomiya et al., 2002) has been developed with a different application in mind. During the implementation and experimentation of overlay, we have developed an efficient deterministic algorithm which we will formally describe below. Our characterization resembles operational semantics more than the set-theoretical characterization given in (Carpenter, 1993), i. e., we describe *how* to compute rather than *what* to compute. But first we will approach the operational semantics by informally introduce the algorithm:

One viewpoint for the characterization of default unification is to unify as much (consistent) information as possible from the defeasible structure into the strict one, e. g., (Carpenter, 1993). Our viewpoint is based on a naive implementation of unification:

> If the types of the covering and background are *compatible* then recursively call overlay with the values of each feature. In case of conflicts, the value of the covering overwrites the value of the background.

However, the compatible condition is too restrictive and limits the function of overlay as shown by the following example taken from the running multimodal dialogue system SmartKom: In the SmartKom domain model, the classes Performance and Broadcast represent film running at the movies and broadcasts on TV respectively. Their common super-type (let us call it BroadcastPerformance) defines common features, such as *Time*. We have the following dialogue excerpt:

(75) **U**: I'd like to go to the movies tonight in Heidelberg

(76) **SK**: Here ($\nearrow$)are the films showing in Heidelberg

(77) **U**: No, there is nothing for me

(78) **U**: what is running on TV?

Now, what is the intended interpretation of utterance 78? There are, of course, several possible interpretations but we favour the one where the user want to see what broadcasts are running on TV *tonight*. However, if we want to use OVERLAY as informally described above, the result of overlaying (the representation of) utterance 78 with (the representation of) utterance 75 would be a structure where the time is omitted. This is the consequence of that we said that the types should be *compatible* and if they are not, the (value of the) covering should overwrite the (value of the) background. This is not what we want! We want the time of the background to show up in the resulting structure. Now, the trick is to transform or *assimilate* the background to the type of the the least upper bound (*LUB*) in the type hierarchy thereby removing features not defined for, in this case, broadcasts, like the town Heidelberg. However, features common to the types of the covering and background are kept.

Our way of thinking is different from that of the classical explanation of default unification, but the result is in fact the same: Unifying as much compatible information from the background into the covering would result in exactly the same structure. Our thinking is based on the requirements posed in a running system which has resulted in an intuitive algorithm.

Operational semantics for Overlay

In what follows, we assume a finite set of types $\mathcal{T} = \{t, t_1, \ldots\}$. The root of the hierarchy is denoted t_{root}. The types are related within a hierarchy in which types can be defined by means of unary inheritance only. That t_i is a supertype of t_j is denoted $t_i \preceq t_j$. Important for the work here is that a feature is introduced once in the hierarchy. This is referred to as *Feature Introduction* (Carpenter, 1992, p. 86). Clearly, (i) the type hierarchy forms a tree and hence (ii) every pair of types has a unique least upper bound (henceforth LUB) which might be t_{root}. We also assume an finite set of typed features $\mathcal{F} = \{f, f_1, \ldots\}$ where f_{t_i} denotes a feature whose value is restricted to the type t_i. Furthermore, we have a (possibly infinite) set of atomic values $\mathcal{A} = \{a, a_1, \ldots\}$ and a set of values $\{v, v_1, \ldots\} \in \mathcal{V} = \mathcal{A} \cup \mathcal{S}$ where $\mathcal{S}$ is the set of typed feature structures. We use the notation $\langle t, \{f_1 : v_1, \ldots, f_n : v_n\}\rangle$[9]

[9] An equivalent notation is $\langle t \wedge \{f_1 : v_1 \wedge \ldots \wedge f_n : v_n\}\rangle$ (see (Krieger, 2001)).

for a TFS where t is the type and $\{f_1 : v_1, \ldots, f_n : v_n\}$ the set of feature-value pairs of the TFS. The notation $f \in t_i$ indicates that the feature f is defined for the type t_i.

Now, we define the *assimilation*[10] of one TFS to another TFS:

Definition 4 (Assimilation)
Let

- $a, b \in \mathcal{S}$ such that $a = \langle t_a, \{f_1 : v_1, \ldots, f_n : v_n\}\rangle$

then, the assimilation of a to b, $a|_b$, is defined as:

$$a|_b := \langle t_{LUB(a)}, \{f_i : v_i, \ldots, f_j : v_j\}\rangle \text{ such that } f_i \in LUB(t_a, t_b) \tag{3.8}$$

□

Informally, the assimilation operation returns a TFS of the type of the more special type in case the types are compatible, i. e., one is subtype of the other, or the type of the $LUB(t_a, t_b)$. Since the type may be generalized, some feature-value pairs might be removed.

Now we can define OVERLAY for typed feature structures:

Definition 5 (OVERLAY)
Let

- $a, b \in \mathcal{S}$ such that

$$a = \langle t_a, \{a_1 : f_1, \ldots, a_n : f_n\}\rangle$$
$$b = \langle t_b, \{b_1 : g_1, \ldots, b_m : g_m\}\rangle$$

- a being called *covering* and b *background*
- $a \sqcup b$ denotes standard unification

then $\text{OVERLAY}(a, b)$ is defined as:

$$\text{OVERLAY}(a, b) := \text{OVERLAY}'(a, b|_a) \tag{3.9}$$

[10] The assimilation function was referred to as “Restriction” in (Alexandersson & Becker, 2001). In this work we have slightly changed the definition and changed the name accordingly. The main reason for this is that the type of the assimilated TFS does not necessarily receive a more general type but an *incompatible* one, and “restriction” might then give a false impression of the functioning of the operation.

$$\textsc{overlay}'(a,b) = \langle t_{a \sqcup b}, \{c_i : h_i \mid$$

$$c_i = a_j = b_k, h_i = \textsc{overlay}(f_j, g_k),\ \text{where } f_j, g_k \in \mathcal{S},\ \text{or} \tag{3.10}$$

$$c_i = a_j = b_k, h_i = f_j,\ \text{where } f_j, g_k \in V,\ \text{or} \tag{3.11}$$

$$c_i = a_j, h_i = f_j, c_i \neq b_k, 1 \leq k \leq m,\ \text{or} \tag{3.12}$$

$$c_i = b_k, h_i = g_k\}\rangle \tag{3.13}$$

□

The first case (3.10) is the recursive step used when the values are typed feature structures. In equation (3.11) the operation turns non-monotonic and covers the case when the values of covering and background are atomic in which case the value in the covering is used. The next case (3.12) is when the feature is absent in background and we use the one in covering. Finally, 3.13 is the case when the feature of the covering has no value: then use the value in background.

Note that OVERLAY is, unlike unification, not a commutative operation. Hence we have $\textsc{overlay}(a,b) \neq \textsc{overlay}(b,a)$, where a and b are two TFS such that $a \neq b$, and *unify*(a,b) would fail. However, if a and b are unifiable, then $\textsc{overlay}(a,b) = \textsc{overlay}(b,a) = \textit{unify}(a,b) = \textit{unify}(b,a)$.

The Scoring Function

OVERLAY is a non-monotonic operation that always succeeds. If our task is to validate a hypothesis against the discourse memory, we will always find a referent regardless how good or bad the hypothesis fits the referent at hand. This was not a problem with the original implementation of the completion algorithm in the VERBMOBIL system, since it succeeded or failed indicating that it was or was not possible to add the new information to this particular discourse object. Our solution to the behaviour of OVERLAY is a scoring function, which computes a metric we can either use for ranking several possible hypotheses, or given a threshold, even disqualify the result of the operation.

Our first attempt to define a scoring function (Alexandersson & Becker, 2001) was based on simple heuristics consisting of a combination of the amount of information contained in the (combined) output and the distance in discourse history. In (Pfleger et al., 2002; Pfleger, 2002) the scoring function has received a more intuitive and formal design. The new scoring function collects a number of values during the overlay operation as indicated below. The numbers in brackets indicate in which equation in Definition 5 (see Page 107) the respective variables are incremented.

co the number of values stemming from the cover (3.11, 3.12)

bg the number of values stemming from the background (3.13)

tc the number of type-clashes (3.8)

cv the number of conflicting values (3.11)

The sum of **co** and **bg** minus the sum of **tc** and **cv** will be weighted by the sum of **co**, **bg**, **tc** and **cv**. This leads to the function 3.14 whose codomain is between -1 and 1.

$$score(co, bg, tc, cv) = \frac{co + bg - (tc + cv)}{co + bg + (tc + cv)} \tag{3.14}$$

The positive extremal indicates two unifiable arguments whereas the negative extremal indicates that all information in the result stems from the cover. All scores between these two extremal indicates that the cover fits the background more or less; the higher the score the better the fit. For the overall discourse processing we have to take into account other, additional metrics, like the distance to the referent etc.

Characterizing Completion in the Overlay frame work

Within the frame work presented, we characterize the completion algorithm used in the running VerbMobil system. Contrary to the overlay algorithm, the completion algorithm is not succeeding for every pair of covering and background, but for those where the two types of the structures are exactly the same. Therefore, the characterization consists of a simplified form of the assimilation operation we call *c-assimilation* which returns the background structure in case the type is the same, or is undefined if they differ. The type hierarchy in VerbMobil is a partial order $\langle \mathcal{T}, \preceq \rangle$ and the c-assimilation operation consists of a simple type check.

Definition 6 C-Assimilation

Let

- $a, b \in \mathcal{S}$
- $t_a, t_b \in \langle \mathcal{T}, \preceq \rangle$

then, the c-assimilation of a to b, $a|||b$, is defined as

$$a|||b := a \text{ iff } t_a = t_b \tag{3.15}$$

□

Since there is no room for different interpretations the scoring function is obsolete.

3.7.7 Discussion

We have developed a robust machinery for tracking propositional content. Since we are not allowed to perform clarification dialogues we use methods much like those in message extraction.

Due to the nature of the statistic recognition method used for recognizing the dialogue acts, it is not possible to enjoy the full flavour of our dialogue acts. Instead, we have grouped 10 of them into a kind of slash group representing dialogue acts which are either not existent in the running system, does not contribute to the negotiation and, finally, are hard or impossible to recognize with a good recall. Furthermore, for guiding the manipulations of the propositional content, we have collapsed the dialogue acts into four sets: *propose*, *feedback*, *elaborate* and *request*. The main reason for this is the conception of our dialogue acts. They are important for distinguishing different acts, e. g., on the surface, during negotiation. However, given this differentiation we can read out the attitude towards our negotiation objects and straight forwardly select the appropriate processing.

A novel feature presented here is the formalization of the completion algorithm—OVERLAY—which together with a scoring function has turned out to be a powerful and generic tool. In fact, it is used as *the* algorithm for a discourse processing in SMARTKOM (Alexandersson & Becker, 2001; Pfleger et al., 2002). Recently, (Pfleger et al., 2002) we have put some effort in the scoring function of the OVERLAY operation. Overlay is indeed too general, and there is a need for a mechanism that can rank the result of the operation. Recall that the completion algorithm presented in section 3.7.3

can fail or succeed. This is not the case for the formalization: OVERLAY *always* succeeds, which further motivates the need for a scoring mechanism. Moreover, as it turns out, a scoring function is essential when it comes to select hypotheses from a parser connected to, e. g., a non-perfect speech recognizer producing several hypotheses. This issue is further discussed in (Pfleger et al., 2002).

The introduction of the *assimilation* operation is motivated by the difference in design of the domain model in SMARTKOM and VERBMOBIL: whereas the domain model in VERBMOBIL uses different values for the role `has_transportation` for distinguishing a *taxi move* from a *move by train*, this would be expressed by specializing the move type into a taxi-move and train-move respectively in SMARTKOM. To still facilitate inheritance of information between structures more or less related to each other, the use of the assimilation operation in the definition of OVERLAY is essential. Finally, in (Alexandersson & Becker, 2003b) we indicate that it is possible to generalize OVERLAY so that we can use it for domain models like the one of VERBMOBIL, where the type system allows for multiple inheritance. The idea is based on the assumption that changing the type system only affects the assimilation operator, leaving OVERLAY untouched.

In (Alexandersson & Becker, 2003a) we have continued to develop the overlay algorithm. In particular, we have extended the operational semantics for domain models employing multiple inheritance.

When it comes to modeling and analysis there are still some unsolved challenges. Particularly, we would like to point at determining scope for the modifiers in time expressions (see section 3.7.3).

Finally, we would like to look back and connect to the definition of pragmatics (section 2.2.1): pragmatics is "*The study of the principles for when (non-anomalous) communicative acts make sense*". Using the OVERLAY operation together with a scoring function as described above, we are (at least), given a certain information state, able to answer the question whether some hypotheses make more sense or not.

3.8 Managing the Intentional Structure

We now turn into the process of constructing and managing the *intentional structure* (see section 2.6). This structure has been designed to be used for two main purposes:

- the computation of the *dialogue phase* used by the transfer module (see section 3.4)

- to arrange the information in the discourse memory in such a way that we can generate minutes

Due to its immaturity, we will give the generation of minutes a rather brief presentation and merely provide some hints on where and how the arrangement of the discourse memory is performed.

As shown in section 2.6 (Figure 2.5, Page 53), the structure is a tree composed by five so-called *levels*. From bottom to top these levels are:

- **The dialogue act level** abstraction of the utterances
- **The dialogue move level** abstraction of the dialogue acts
- **The dialogue game level** abstraction of the moves
- **The dialogue phase level** abstraction of the games
- **The dialogue level** abstraction of the phases

We use a combination of techniques presented earlier as well a new component—the plan processor—presented below for the construction and maintenance of the intentional structure. The plan processor interprets plan operators and the structure of the plan operators form a tree which we view as the intentional structure. A striking experience made early in the project was that it is very hard to hand-craft plan operators for building the structure covering more than, approximately 20 dialogues. Given good structures—thus a good coverage—for those 20 dialogues, the same set of operators failed for a big part of the next 10. Therefore we started to investigate methods from the area of machine learning. To enhance robustness, we would like to highlight the following solutions:

- For the *processing* of moves, i. e., building the moves structure, we utilize the close relationship between rules in a context free grammar (CFG) and plan operators. Instead of hand-crafting the plan operators we learn a (probabilistic) CFG from an corpus annotated with dialogue acts and moves (see section 3.8.2).
- For the *classification* of moves we have used n-gram models. Instead of using words as tokens for recognizing dialogue acts, we use dialogue acts as tokens for recognizing moves. A well-established fact is that the prediction power of n-gram models is superior to that of stochastic context free grammars; a fact that has been verified in our own experiences. This is described in section 3.5.1.

A challenging task is the segmentation of turns into moves. Our solution is to use the forward/backward looking aspects of the dialogue acts (see section 3.8.4). In section 3.8.7 we describe how to build the rest of the structure (the games, phase and dialogue level). There, we utilize the plan processor as a (top down) parser and parse the moves with hand written plan operators. These plan operators reflect the modeling of games as described in section 2.6.3 and the three phase nature of our cooperative negotiation dialogues: opening, negotiation and closing. However, we start by presenting the plan processor.

3.8.1 The plan processor

The plan processor, e. g., (Alexandersson et al., 1997), is an interpreter interpreting plan operators much in the same way a Prolog interpreter interprets Prolog programs. We introduce the plan operators in the next section and show how they are related to other relevant work utilizing plan operators. For the work presented in this chapter, an important observation is that clauses in a Prolog program on the one hand and rules in a context free grammar on the other are closely related (Deransart & Małuszyński, 1993). This observation opens up the way for making use of well-known efficient parsing methods as well as machine learning methods for the construction of the intentional structure on the one hand and automatic acquisition of plan operators on the other.

Next, we present the syntax and properties of plan operators and how our definition is related to other plan operators in the literature as well as Prolog programs. We also briefly discuss the interpretation algorithm.

Plan Operators

We define plan operators using the syntax as shown in figure 3.5. We denote expressions bracketed within "`[`" and "`]`" as optional. Expressions delimited with "`|`", e. g., "`a | b`" denote "`a`" (exclusive) or "`b`".

Every operator has a `:name` (a symbol) and a `:goal` which is a lisp expression, i. e., a possibly dotted lisp list or a symbol. `<lisp-code>` and `<goal>` are also lisp code, but differ in that the symbols `:and` `:or`, `:sor`[11], `:xor` and `:optional` have a different meaning. They are interpreted in the goals during the compilation of the plan operators, and factored out possibly

[11] `:sor` is short for sequential-or. This is interpreted as or with the difference that all permutations of a certain combination of the arguments appear only once, namely in the order they appear, e. g., `(:sor a b)` means one of `(a)` `(b)` or `(a b)`, but not `(b a)`.

```
<plan> ::=     (def<what> <name>
               :doc <string>
               :goal <goal>
               [:constraints <lisp-code>]
               [:pre-actions <lisp-code>
               :undo-pre-actions <lisp-code>]
               [:actions <lisp-code>
               :undo-actions <lisp-code>]
               :subgoals <goal>
               [:prob <integer>]
               ).
<defwhat> ::=  defoperator | deffact | definvisible
```

Figure 3.5: Plan Operator Syntax

yielding more plan operators. In lisp code these symbols are regarded as ordinary (common-lisp) keywords.[12] Symbols prefixed with a question mark "`?`", e. g., `?foo`, are interpreted as variables. Variables prefixed with "`!`" are used in, e. g., the constraints, to compute the value of that variable. Variables are either unbound or bound during interpretation of the plan operators.

Our plan operators resemble Prolog clauses with some additional decorations, e. g., `:constraints,` `:actions`, and `:prob`. A Prolog clause is composed by *atoms*.

$$H \leftarrow B_1, \ldots, B_n \tag{3.16}$$

One can think of atoms as an expression of the form $p(t_1, \ldots, t_n)$ where p is an n-ary predicate symbol, and t_i terms. In 3.16, H is usually called *head* and $B_1, \ldots, B_n$ the *body.* A Prolog clause with an empty body is usually referred to as a *fact.* For our plan operators, we will refer to the head as *left hand side* or *goal*, and to the body as *right hand side* or *subgoals.* The main syntactic difference is that goals and subgoals of a plan operator are written as lisp lists, where the predicate in a Prolog clause is written as terms. The predicate symbol in Prolog correspond to the first symbol in the list.

[12] Readers not familiar with the syntax and semantics of Common-Lisp are referred to (Steele, 1984).

Interpreting the Plan Operators

For the interpretation of plan operators, we use a standard Prolog top-down left-to-right expansion algorithm with chronological backtracking. We use Prolog-like unification (Charniak, Riesbeck, McDermott, & Meehan, 1987) to match goals and subgoals. If the `:constraints` evaluates to `nil`[13] the interpretation algorithm backtracks. If the interpreter enters an operator the `pre-actions` are evaluated whereas the `:actions` are evaluated when all sub-goals have been successfully interpreted. Both `:actions` and `:pre-actions` are "undone" chronologically during backtracking by evaluating the `:undo-actions` when entering the operator and the `:undo-pre-actions` when leaving the operator. The reason for this complicated construction is due to the environment in which the plan processor works: The consequence of solving all sub-goals of a plan operator, i. e., evaluating the actions, might be to *destructively* write the dialogue phase into the sequence memory. However, if this particular operator is backtracked away, the side effects have to be undone. Finally, `:prob` is used for sorting competing operators.

To increase efficiency, the plan processor additionally facilitates the possibility of dynamically saving a derivation, i. e., tree, during backtracking. Finally, the plan processor is implemented in the Common List Object System (CLOS) which, in an easy and convenient way, makes it possible to specialize parts of the plan processor.

To summarize, we have a flexible and powerful tool which can be used for several tasks, such as:

Parsing/Plan Recognition There is a close connection between Prolog like rules and context-free grammars (Deransart & Małuszyński, 1993; Boye, 1996). We can thus view a plan operator as a rule in a CFG. We can also compile a set of plan recognition operators, e. g., (Kautz, 1987) into a CFG and parse the dialogues (Vilain, 1990) using efficient top-down-parsing strategies, e. g., (Early, 1970; Stolcke, 1995).

Text Generation (Moore & Paris, 1993) proposes plan operators for the generation of coherent multi-sentential text. As indicated above there is a projection from their plan operator language into ours.

Prolog Programming Although the full flavour of Prolog is not implemented,[14] "smaller" Prolog programs can be implemented. This opens

[13] `NIL` is the lisp-equivalent to false.

[14] diff-lists, cut and other nice Prolog features have been omitted in our implementation.

the door for a wide variety of applications, including constraint solving, rule expansion etc.

3.8.2 Acquiring Plan Operators and Language Models

Early in the project we experienced the lack of robustness when writing plan operators by hand for the parsing of dialogues. Once we succeeded in analyzing a couple of dialogues, the next couple always contained new phenomena not covered by our experiences up to that point. This forced us to look for more robust methods for the construction of the intentional structure. Several methods - both supervised and un-supervised - for the acquisition of probabilistic language models from, possibly annotated, corpora have been proposed, e. g., (Stolcke, 1994a; Chen, 1996).

The *Boogie*[15] system (Stolcke, 1994b) provides a Bayesian frame work for, e. g., deriving probabilistic context free grammars (PCFG) and n-gram language models. We have used this system for learning PCFGs for the moves layer using our corpus annotated with dialogue acts, moves and games.

For the acquisition of statistical knowledge, we partition our corpus of dialogue annotated with dialogue acts and moves. Each partition is a set of examples for each move. We use these partitions to derive:

- **Probabilistic context free grammars** As indicated above, we use the Boogie system (Stolcke, 1994b). These are transformed to our plan operators syntax (see section 3.8.1).

- **N-gram models** Here we use the work bench and methods described in, e. g., (Reithinger & Klesen, 1997).

The grammar in figure 3.6 is one sample grammar derived for the greet move.

3.8.3 Adapting the Plan Processor for building the Intentional Structure

Before we can use the plan processor for building the intentional structure we have to extend it a bit. First, we use the tree imposed by the plan operators much like the back-bone in a context-free grammar, i. e., we view the derivation as a parse tree.

The first extension is based on the observation that it is convenient to perform actions or computation "in between" the plan operators forming

[15] Bayesian Object-oriented Grammar Induction and Estimation

```
s-10455  ->  (1.0) s-10456
s-10456  ->  (9.8e-3) s-10457 s-10458 s-10459 |
             (0.775) s-10458|
             (0.216) s-10458 s-10459
s-10457  ->  (1) accept
s-10458  ->  (1.4e-2) feedback |
             (0.986) greet
s-10459  ->  (4.76e-2) inform |
             (0.190) introduce |
             (0.762) politeness_formula
```

Figure 3.6: The PCFG for the greet move. The numbers in brackets are the probabilities produced by the Boogie system. The root of the grammar i the top-most rule (`s-10455`).

the tree. We, therefore, introduce specialized plan operators which are not "visible" in the intentional structure. Such an operator is defined with the keyword `definvisible` in figure 3.5 on 114. Additionally, the plan processor is extended with a dot similar with the dot in early parsing - consuming an input segment causes the dot to be moved ahead one step if possible whereas the dot is moved backwards during backtracking. During processing of invisible plan operators, the moving of the dot is switched off. In this way we can perform computations using the same language without messing up our target structure. An example of the usefulness of this functionality is the manipulation of the tree context (see Section 3.8.8).

By introducing a subclass of facts called *leaf operators* or *dialogue act operators* (defined using the constructor `defleaf`) we connect the intentional structure to the dialogue acts. During interpretation, the argument of the `:leaf` is unified with the content of the dialogue act slot in the current segment. Figure 3.7 on page 118 shows two examples of leaf-operators. The code in the `:actions` slot of the `SUGGEST-LEAF` operator is responsible for, e. g., setting the dialogue phase in the sequence memory. First the phase is looked up in the tree context. If not set, the negotiation phase is entered in the tree context and finally written into the sequence memory. The actions of the second operator marks this segment as irrelevant for the minutes—deliberations are always irrelevant—and propagates the tree context by unifying the input with the output variable.

Finally, the interpretation of the plan operators for the intentional structure is continued as long as the current segment pointed at by the abovementioned pointer has a successor. If not, the interpretation algorithm stops.

```
(defleaf SUGGEST-LEAF
    :doc ""
    :goal (domain-dependent SUGGEST-LEAF ?in ?out)
    :actions (push-dialogue-phase-and-set-context-actions
             :default *phase-negotiation*
             ?in ?out)
    :undo-actions (nop)
    :leaf SUGGEST
)

(defleaf DELIBERATE-LPO
    :goal (domain-dependent DELIBERATE ?in ?out)
    :actions (progn (unify! ?in ?out)
                   (minutes-irrelevant!
                    (current-segment *planner*)))
    :undo-actions (nop)
    :leaf  DELIBERATE)
```

Figure 3.7: Two leaf operators.

3.8.4 Processing Flow of the Intentional Structure

Whereas the segmentation of turns into utterances or segments relies on prosodic information, there is no prosodic information available for the segmentation of turns into moves. Instead we use the forward/backward looking aspect of the utterances as described in section 2.6.2. The algorithm is straight forward. We assume that every turn has either a backward, a forward or a backward and a forward looking aspect (see Section 2.3.2). We also say that some dialogue acts, such as SUGGEST, REQUEST_SUGGEST and INIT are *forward looking*, whereas dialogue acts such as REJECT and ACCEPT are *backward looking*. Technically, there are also four other "lookings": *greet*, *bye*, *commit* and *close* lookings—all corresponding to one of the phases as described in section 2.6. All dialogue acts not belonging to one of these classes are called *neutral looking* dialogue acts. These dialogue acts are neither providing strong enough evidence for forward nor backward looking aspects. The corresponding dialogue acts are listed in table 3.8.

While encountering a dialogue act with some other looking than neutral looking, this part of the turn is regarded as belonging to one segment. As soon as the looking changes from, e. g., backward to forward, a segmentation is performed. We clarify with an example. Suppose we have the following

Table 3.8: The different lookings and their corresponding dialogue acts.

Looking	**Dialogue Acts**
Forward	REQUEST REQUEST_COMMENT REQUEST_SUGGEST REQUEST_COMMIT REQUEST_CLARIFY SUGGEST INIT
Backward	REJECT EXPLAINED_REJECT ACCEPT FEEDBACK FEEDBACK_POSITIVE FEEDBACK_NEGATIVE
Greet	GREET INTRODUCE
Bye	BYE
Commit	COMMIT
Close	CLOSE
Neutral	OFFER CONFIRM INFORM DIGRESS DEVIATE_SCENARIO REFER_TO_SETTING EXCLUDE GIVE_REASON CLARIFY INFORM_FEATURE DEFER SIGNAL_NON_UNDERSTANDING BACKCHANNEL POLITENESS_FORMULA DELIBERATE THANK NOT_CLASSIFIABLE

turn to process:[16]

(79) ist zwar ein bißchen früh (GIVE_REASON)
is though a bit early
it is a bit early though

(80) aber können wir machen normales (ACCEPT)
but could we do normal
but we could do that

(81) dann treffen zum Früstück (SUGGEST)
then meet to breakfast
let us meet at breakfast

The first segment (79) is annotated with a neutral looking dialogue act. Segment 80, however with a backward looking dialogue act. The turn up until this point is still regarded as one move - response. Not until we get to the third segment, does the looking change from backward to forward, and we segment the turn between segment 80 and 81.

Finally, for longer turns, i.e., 5 and more segments, we use heuristics to keep the number of segments down. Due to, for example, imperfect recognition of dialogue acts and false segmentation of the spoken input the

[16] Taken from one of our "dialogue of the week".

looking might change more than one time within the turn sometimes giving birth to unpredictable structures.

3.8.5 Recognizing moves

Once the turn is segmented, our next task is to recognize the move for the segment. We have tried two ways of recognizing the moves. The first is based on the probability of the acquired CFGs, and the second based on n-gram models. As a basis for our experiments we use a part of the VerbMobil corpus annotated with dialogue acts and moves which consists of 277 dialogues. For this experiment we have randomly split the data into four partitions where each partition consists of 3 disjunct sets: 70% for training, a validation set (20%) for adjusting the parameters (see below), and 10% for testing. We evaluate our approach in four experiments to show with a variety of data how well the approach performs. On average, the probability of the grammar yielded a hit rate of 41.17%. Our n-gram models (including predictions) reached 72.02% (see (Alexandersson & Reithinger, 1997) for similar experiments). Worth noting is that the PCFGs has a coverage of about 90% on unseen moves.

To our knowledge, there is very little work in this direction. (Jurafsky et al., 1995) describes an experiment where a SCFG in combination with a bi-gram language model is used to reduce the word error rate for speech recognition. While the word error rate is actually reduced it is not clear from the description how well the coverage of the grammar is.

3.8.6 Building the moves structure

The plan operators for the moves structure can be divided into four sets containing:

1. plan operators compiled from the dialogue act hierarchy in Figure 2.6,

2. plan operators derived as described in Section 3.8.2 for building the move structure,

3. plan operators manually written for building the three top most layers, and

4. plan operators manually written for, e. g., maintaining the tree context (see Section 3.8.8)

The distribution of plan operators are given in table 3.9. There are in total 473 plan operators where 303 have been acquired from the annotated corpus, 78 have been compiled from the dialogue hierarchy, and 92 are hand written.

Table 3.9: The distribution of plan operators to move classes. There are a total of 303 semi-automatically acquired operators for the move classes. The class "domaindependent" is the result of compiling the dialogue act hierarchy into operators, whereas "top" consists of handwritten operators for games, phases, and top layer. Finally, "misc" contains plan operators for, e. g., maintaining the tree context.

greet	10
bye	17
init	20
close	23
transfer-initiative	14
initiative	57
response	9
response+initiative	35
confirm	19
unknown	15
signal_non_understanding	2
skip	14
garbage	3

clarify-answer	18
clarify-query	16
request-commit	5
commit	22
request-describe	9
describe	14
Σ :	303
domaindependent	78
top	66
misc	23
Σ: 303 + 78 + 66 + 26 =	473

Robustness

As indicated above, the learned plan operators have a coverage of around 90%. If we want to build the intentional structure out of one big parse tree, failure in building the structure for one particular move implicates failure of the structure as a whole. Therefore, we instantiate a new plan processor for each move which receives the task of building the move structure for this particular move. Such a plan processor is called a *move planner* – the plan processor responsible for the upper layers is from now on called the *top planner*. If a move planner fails, the move structure is built using a set of fall-back plan operators capable of building a left recursive structure for any sequence of dialogue acts. This guarantees a complete tree although the structure might be unintuitive. The processing starts with the prediction of the move type using language models possibly proceeded with a segmentation. Given the predicted move, we start by expanding the plan tree with the set of plan operators for the predicted move. The next step is to activate the top planner. We define a successor and predecessor relation on the move planners (in the same way the segments of the sequence memory) thus allowing the leaves of the higher level structure (see the next Section) to connect to the sequence of move planners as a move planner connects to the sequence of segments in the sequence memory.

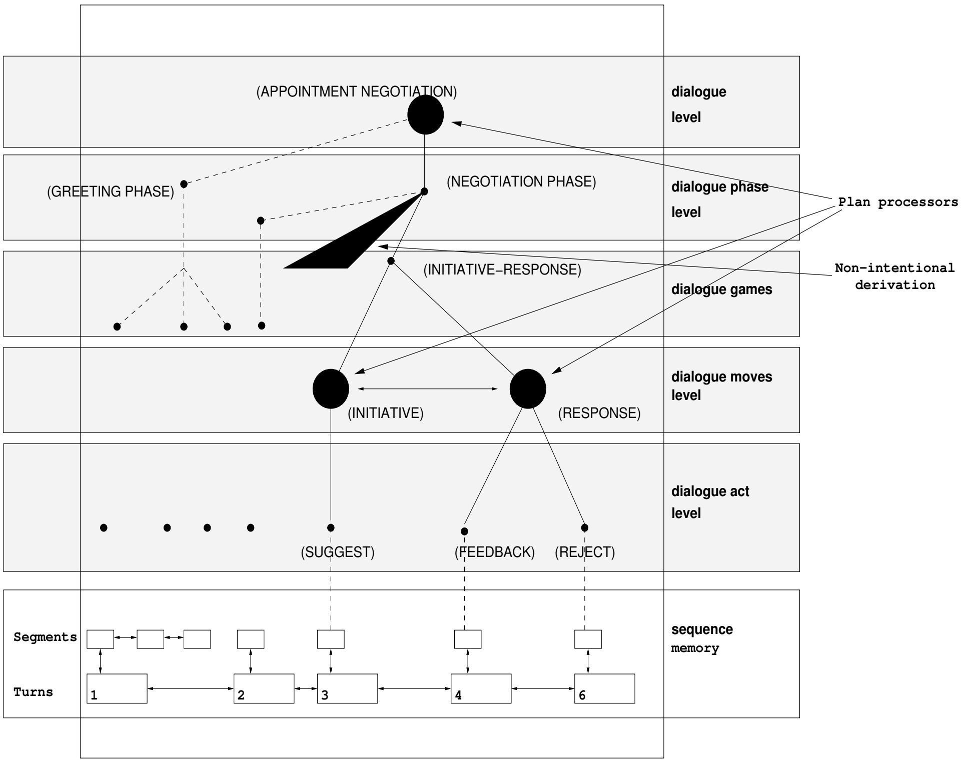


Figure 3.8: Processing the intentional structure.

3.8.7 Building the rest of the Structure

The three top most layers of the intentional structure—the dialogue, the dialogue phase, and the dialogue games layer—are built using hand-crafted plan operators modeling dialogue games and phases as described in section

2.6.3. This planner we call the *top planner*. Starting at the root, the complete negotiation dialogue is modeled with, e. g., the operators depicted in figure 3.9. The first one—"`ONE-APPOINTMENT-OPERATOR-complete`"—is the most prominent root of the tree. The operator named `NEGOTIATION-PHASE--OPERATOR` is used to model sequences of I-R games. The (sub-) goal `(ITER TOP NEGOTIATION GAME)` unifies with a set of operators allowing for an arbitrary number of, in this case, negotiation games. Some simplified operators are depicted in figure 3.10.

```
(defplan ONE-APPOINTMENT-OPERATOR-complete
  :doc "Complete negotiation"
  :goal (TOP ONE-APPOINTMENT)
  :subgoals (:sequence
             (TOP GREETING PHASE)
             (TOP NEGOTIATION PHASE)
             (TOP CLOSING PHASE))
  )

(defplan NEGOTIATION-PHASE-OPERATOR
  :doc "Iteration of I-R games"
  :goal (TOP NEGOTIATION PHASE)
  :subgoals (:sequence
              (ITER TOP NEGOTIATION GAME)))
```

Figure 3.9: Plan operators for *i*) a complete negotiation dialogue and *ii*) arbitrary number of I-R games.

The most prominent game is the *negotiation* game for which some of the plan operators are depicted in figure 3.10.

The leaf operators of this part of the intentional structure are connected to the move planners described in section 3.8.6 by unifying the value of the `:leaf` slot with the value of the `:goal` slot of the root of the move plan.

3.8.8 Setting the dialogue phase

The dialogue phase information is managed by the plan operators within the dialogue phase level in combination with the operators for the dialogue act level. The solution is based on three simple and robust techniques:

- **tree context** The goal and subgoals of the plan operators depicted in, e. g., figure 3.10 are extended with two arguments—one input and one output variable—which carry a *tree context* around in the tree.

```
(defplan Nego-I-R
  :doc "I-R pair"
  :goal (TOP NEGOTIATION GAME)
  :subgoals (:sequence (TOP NEGOTIATION INITIATIVE)
                       (ITER TOP NEGOTIATION RESPONSE)))

(defplan Nego-T-I-R
  :doc "Complete I-R pair"
  :goal (TOP NEGOTIATION GAME)
  :subgoals (:sequence (TOP NEGOTIATION TRANSFER-INITIATIVE)
                       (TOP NEGOTIATION INITIATIVE)
                       (ITER TOP NEGOTIATION RESPONSE)))

(defplan Nego-I-R+CD
  :doc "I-R pair with embedded clarification dialogue"
  :goal (TOP NEGOTIATION GAME)
  :subgoals (:sequence (TOP NEGOTIATION INITIATIVE)
                       (ITER TOP NEGOTIATION CL-GAME)
                       (ITER TOP NEGOTIATION RESPONSE)))
```

Figure 3.10: Plan operators for the negotiation game

The tree context can be viewed as a frame or an association list. One of its features is *dialogue-phase*.

- **dialogue phase and dialogue operators** The dialogue and dialogue phase operators are decorated with non-intentional goals (see Section 3.8.3) which set and change the dialogue phase in the tree context.
- **dialogue act operators** The dialogue act operators have an action which is performing two tasks:
    1. in case there is no phase available,[17] set the phase to, e. g., NEGOTIATION for an operator responsible for the dialogue act SUGGEST, and add the phase to the context.
    2. destructively write the dialogue phase in the dialogue sequence memory

The manipulation of the tree context is performed by a set of non-intentional plan operators which allow for, e. g., changing, adding, removing a keyword

[17] It sometimes happened that – due to erroneously recognized dialogue acts or false segmentation – the tree context did not contain a value for the dialogue phase.

or its value. Then, we extend the plan operators in figure 3.9 as depicted in figure 3.11.

```
(defplan ONE-APPOINTMENT-OPERATOR-complete
  :doc "Complete negotiation"
  :goal (top one-appointment ?IN ?OUT)
  :subgoals (:sequence
             (tree-context (?in ?out1) set (dialogue-phase
                                            *phase-hello*))
             (top greeting phase ?out1 ?out2)
             (tree-context (?out2 ?out3) change (dialogue-phase
                                                 *phase-negotiation*))
             (top negotiation phase ?out2 ?out3)
             (tree-context (?out3 ?out4) change (dialogue-phase
                                                 *phase-bye*))
             (top closing phase ?out4 ?out))
  )
```

Figure 3.11: A plan operator for a complete negotiation dialogue extended with the tree context. In the `:goal`, the variable `?IN` is the input variable and the `?OUT` is the output variable.

3.8.9 Discussion

One of our main motivations for the intentional structure is pointed out by (Buschbeck-Wolf, 1997). She indicates that not just local context is important for improving the translation. Our intentional structure is one of the contributors for enhancement of the translation by determining the dialogue phase.

For the implementation, there are several results and projects which have inspired our solution to constructing and maintaining the intentional structure. (Vilain, 1990) describes how plan recognition can be viewed as parsing by compiling a library of a formal plan recognition model to a context free grammar. One of the motivation factors for this is processing efficiency. (Scha & Polanyi, 1988) presents an augmented CFG for parsing dialogues. Whereas both results are theoretical and appealing, they rely on hand crafted models. Our own experiences show, however, that such models are inflexible and that processing based on such approaches is not very robust outside the scope of prototypical dialogues. Therefore we used machine learning-methods to learn the structures, i. e., plan operators, from an annotated corpus. Also, we used techniques from speech recognition—n-gram models—for the recognition of the dialogue moves. To further enhance robustness, the intentional structure is implemented using multiple instances of our plan processor.

Other usage of higher level structures is provided by (Poesio & Mikheev, 1998) where the authors investigated whether higher level, hierarchical structures—conversational games—affect the *prediction* of dialogue acts and conversational moves using the map task corpus (see 2.3.4). Four experiments were performed. In which, in the first one, only a bi-gram model was used. The result was 38.6% for the first being correct and 52% for one of the two first being correctly predicted. In the second test, the information about the current game and the position in the game was taken into account. The result increased to 50, 63% and 67, 07% respectively. By including the speaker direction the accuracy increased to 54% for the first move being correctly predicted. In the last experiment, as suggested in (Poesio & Traum, 1997), the dialogue control moves—ACKNOWLEDGE and CLARIFY—were separated from task-oriented moves yielding 57, 2% for the first hypothesis correct and 72, 3% for one of the two first being correct predicted.

(Qu, Rose, & Di Eugenio, 1996) describes how contextual information given by a plan processor (Lambert, 1993) has the potential to be useful for resolving ambiguity produced by the parser and thereby improving translation accuracy in the *Enthusiast* Spanish-to-English translation system. The positive effect is described as:

> "...we can achieve an improvement of 13% with the genetic programming approach and an improvement of 2.5% with the neural net approach over the parser's non-context based statistical disambiguation technique."

However, the effect of *cumulative error*, i. e., resolving an ambiguity based on a prior false assumption, makes the recognition drop considerably.

(Chu-Carroll, 1998) proposes a model for the *recognition* of dialogue acts taking not just previous dialogue acts into account but additional information, like sentence mood, and higher level structures in the spirit of (Grosz & Sidner, 1986). The recognition of the dialogue act varied from about 30% to 50% for the first dialogue act depending on what information was used. The results are hard to compare with those of (Poesio & Mikheev, 1998) since the higher-level structures were only used during training.

To conclude, there are pros and cons of incorporating higher-level structures into the recognition of, e. g., dialogue acts. It is possible to improve the recognition of dialogue acts. Several results also show the potential of utilizing such information. However, given our approach, which resemble several of the above-mentioned approaches in several respects, there are reasons against such an adventure:

- The mutual dependencies between, e. g., dialogue acts and games may cause the recognition to "deviate", i. e., a recognition error on one level effects the recognition negatively on the other level. This is indicated in (Qu et al., 1996), where the cumulative error problem seems to degrade the system performance instead of enhance it.
- It is hard even for human annotators to annotate a dialogue model reliably (Carletta et al., 1997).

Under the processing circumstances in VerbMobil, we can just hope for an approximation of the intended model. Future work will have to show whether our approach is useful for the recognition of dialogue acts for the VerbMobil scenario.

3.9 Conclusion

This chapter describes the different processing techniques used by the dialogue module in VerbMobil. To be able to process massive amounts of previously unseen data, we have used the combination of more traditional hand-coded knowledge, on the one hand, and different robust methods from the area of speech processing and machine-learning on the other hand. Instead of investing our time with the hand-coding of knowledge sources, our focus has been on the development of (theoretical) models and their corresponding annotation schemata. We have then annotated and used the corpus for training our models or acquisition of, in this case, context free grammars and n-gram models. Also, our approaches have been evaluated against our corpus (see section 3.8.5).

One major contribution of this thesis is the formalization of the completion operation – overlay – which is used not only to add contextual information to the propositional content of a new utterance, but also to overwrite conflicting information. We believe the overlay operation to be the basic operation for managing the discourse state in dialogue systems basing their processing on frame-based formalisms. Recent investigations indicate that overlay can handle phenomena like gapping, one-anaphora and verb-anaphora.

Our approach to discourse modeling differs to those of many other man–machine dialogue systems. Instead of relying on *one* fragile processing technique, we combine several methods, yielding a robust system with a broad coverage. In the next chapter, we will evaluate parts of the content of our discourse memory by generating summaries. There, a carefully selected subset of the memory is used to mirror the result of the negotiation dialogues.

Chapter 4

Generating Multilingual Summaries

There are two ways of viewing the content of this chapter. One is based on the perception that summary generation and evaluation are two distinct disciplines. This is probably true and could hence be used as an argument for making two chapters out of this one. There is, however, a good reason for presenting these two topics under one umbrella. The two above-mentioned disciplines are, in fact, tightly coupled; the main reason being that nowadays, research in computational linguistics in general and, in particular in the field of summarization, has progressed such that research always includes an evaluation. This chapter contributes to both disciplines. So, here we go:

This chapter contains two parts. The first part, consisting of the sections 4.1–4.4, describes a functionality of the running VerbMobil system we call *Summary generation*. The key idea behind the the generation of summaries is to provide the users with a document describing the result of the negotiation. The generation itself is based on part of the contents of the dialogue memory, namely the most recent specific accepted negotiation objects which contain all objects in the discourse that the speakers agreed upon. In the running system, one related additional functionality has been implemented, namely the *dialogue minutes*. The minutes rather mirror the flow of the dialogue encompassing the more salient parts of the negotiation. For generating the summaries and the dialogue minutes a generator called the *summary generator* has been developed which—unlike most other prominent work in the area of natural language generation—processes the input structure in a bottom-up fashion. It has been possible to realize the summary generation functionality within VerbMobil because our approach to representation and the processing thereof. In contrast to other translation tracks, e. g., substring-based and example-based translation, the approach pursued by the dialogue act based translation track—based on dialogue act and instances of our domain model—allows for a robust and a comparatively straight forward management of negotiated discourse entities. The task of collecting and managing this information is a challenge of its own, but during the course of the dialogue it mainly serves the tasks of choosing the resolving anaphoric and interpreting elliptical references and utterances, tasks that have to be supported by any dialogue manager of a dialogue system. At the end of the negotiation, the content of the dialogue memory in VerbMobil mirrors the accepted and rejected negotiable objects, bundled in such a way that it is well suitable for generation. Additional linguistic knowledge, i. e., the semantic database and the VIT, provide a clean interface to existing infrastructure within the VerbMobil system, e. g., a semantic transfer module and a multilingual natural language generator. We can therefore generate well-formed documents in any language implemented in the system not based on copying salient user contributions but on generation from instances of our domain model. The final document is thus a compact well-formed representation of the negotiation result.

The second part, consisting of section 4.5, contains a proper evaluation of the summarization functionality. We have evaluated the summary generation by comparing the content of the summaries for both mono-lingual as well as multi-lingual negotiation dialogues with a gold standard, i. e., hand-made summaries. Evaluations based on hand-transcribed dialogues and dialogues using one of our speech recognizers are provided. We additionally define a new concept—*confabulation*—which captures a certain kind of error a summary system produces when processing, e. g., spontaneous spoken language. We distinguish between erroneously *selected* objects and "made up" or "hallucinated" objects stemming from any step in the processing chain. Typical examples of such errors are recognition errors during speech recognition, or inference errors.

Most of the material in this chapter has been presented earlier although the presentation here is more detailed. See (Alexandersson, Poller, Kipp, & Engel, 2000; Alexandersson & Poller, 2000, 1998; Reithinger et al., 2000) for prior publications on summarization, and (Alexandersson & Poller, 2000, 1998) for the generation of dialogue minutes. Section 4.5, though, has not yet been published.

We start by an introduction to the field of summarization (section 4.1) followed by a summary of the field automatic summarization (section 4.2). Prior to section 4.4, describing the summary generator, we recapitulate relevant generation issues in section 4.3. Section 4.5 describes the Evaluation and the concept of confabulation and we conclude the chapter in section 4.6.

4.1 Summarization

What is a summary? Mani (Mani, 2000-2001) gives a nice characterization:

> "The goal of automatic summarization is to take an information source, extract content from it, and present the most important content to the user in a condensed form and in a manner sensitive to the user's or applications's needs."

Previous work on summarization often use newspaper text as a source (McKeown & Radev, 1995) but spoken language has also been used, e. g., (Zechner, 2001b, 2002). Other approaches pursue a more narrow domain, e. g., scientific articles (Teufel & Moens, 2000), whereas other head for unrestricted text (Marcu, 1998).

What is a summary in VerbMobil? Contrary to the summarization of, e. g., newspaper text in an open domain where the *salient* parts of the text are collected, summarization in VerbMobil consists of recapitulating the negotiated objects agreed upon by both interlocutors. For text summarization one important feature is the reduction in complexity or compression rate and thus in length. Indeed, the reduction of length is present in our scenario too, but merely as a side effect: instead of focusing on reduction of text size, we reduce the dialogue to mirror the final agreement of the negotiation.

Another difference between open domain text summarization and our work here lies in the implementation. For more general approaches, there are usually no, very poor or shallow models of the domain. In VerbMobil, in contrast, there is a domain model (see section 2.7) which we use for modeling content and attitude towards the content in form of dialogue acts—see section 2.6.1. However, the task of *generating* the summaries differs from that of, e. g., response generation in a dialogue system in that the generation of summaries requires converting *all* of the content of the knowledge base, e. g., (Marcu, 1997) in contrast to those techniques advocated by, e. g., (Moore & Paris, 1993; Hitzeman, Mellish, & Oberlander, 1997).

Figure 4.1 shows a graphical characterization of "The Summary Machine" as presented in (Hovy & Marcu, 1998). There, the summarizer consists of four modules:

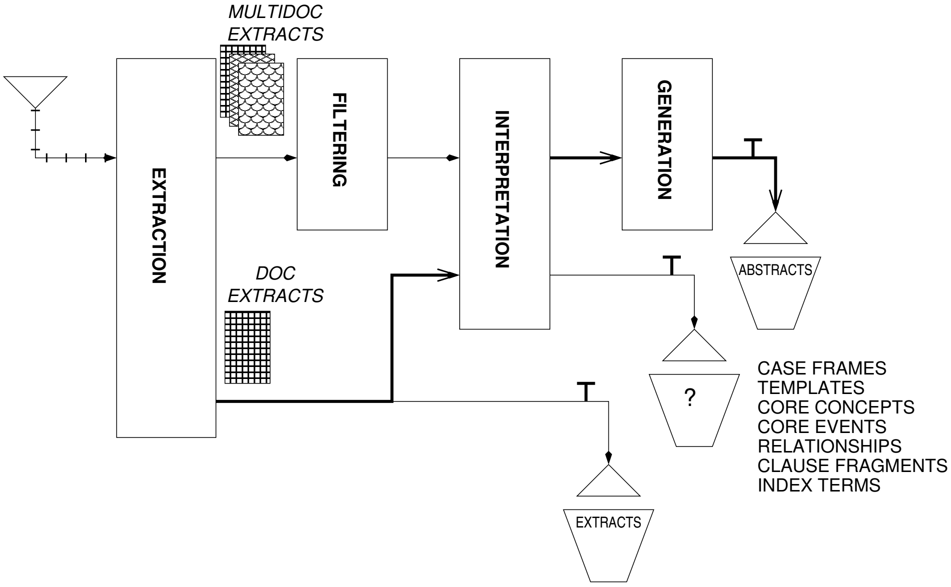


Figure 4.1: The Summary Machine. The dotted lines indicate how the processing in VerbMobil relates to the summary machine. The EXTRACTION module corresponds to the *syndialog* module and the INTERPRETATION module to the *dialogue* module of VerbMobil. Finally, the module corresponding to the GENERATION module is presented in this chapter.

- **Extraction** The "interesting" parts of the input source are extracted. For some systems, extraction is merely selecting parts of the source, but extraction may stand for filling templates, e. g., (McKeown & Radev, 1995).

- **Filtering** Some extraction methods might return too much redundant information. To cope with this, different filtering techniques are used, e. g., (Zechner, 2001b).

- **Interpretation** For systems working with higher level representations of the source, the extracted and eventually filtered information is then interpreted in context, e. g., (McKeown & Radev, 1995).

- **Generation** Abstract representation is realized into text, e. g., (McKeown & Radev, 1995)

In the first phase of VerbMobil, two modules comparable to two of those of the summary machine were developed:

- the *syndialog* module uses extraction methods to output utterances represented as dialogue acts and propositional content.
- the *dialogue* module can be viewed as the interpretation module. The task of the dialogue module is to interpret the output from the syndialog module in context. Amongst other things, the discourse memory keeps track of what has been agreed upon. This information is accessible at any time during the course of the dialogue.

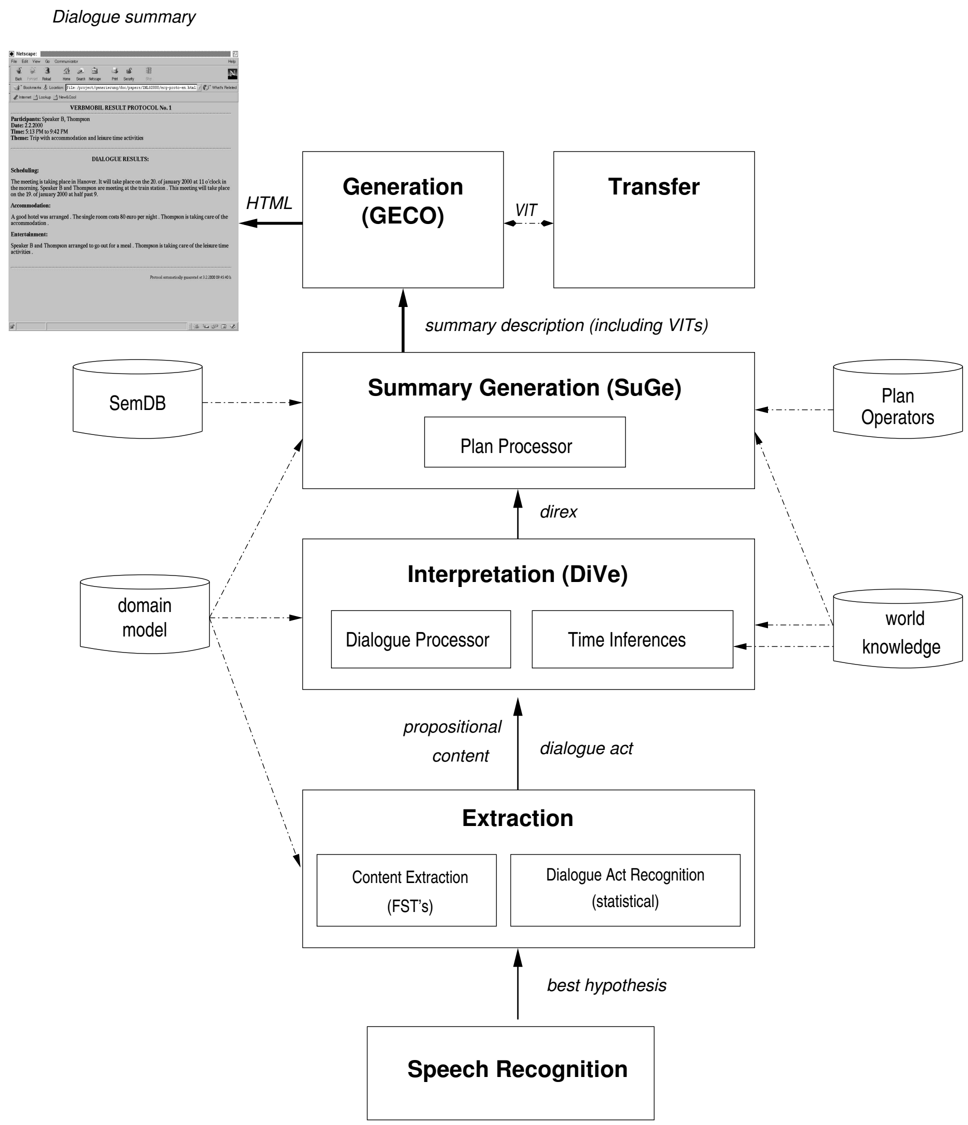
To obtain a summarizer within VerbMobil that resembles the allegation in figure 4.1 a module capable of taking us from the instances of our domain model to text was missing. At the same time we strived to utilize as many of the already existing modules of VerbMobil as possible. Figure 4.2 shows the architecture of our solution. We assume that it would suffice to implement a generator (henceforth the summary generator) taking the agreed-upon negotiation objects, i. e., the most specific accepted negotiation objects from the discourse memory, convert them into German VITs (Dorna, 2000). Then, we can utilize the existing generator GECO (Becker, Kilger, Lopez, & Poller, 2000b) to realize the VITs. Of course, the GECO generator will have to be extended with a formatting functionality. Utilizing the semantic transfer component (Emele et al., 2000) warrants the generation of summaries in other VerbMobil languages.

The motivations behind summaries and indeed minutes in the VerbMobil scenario are manyfold, and we give two of the main reasons here:

- The user has a document describing, on the one hand, what was agreed upon in the case of a summary, or, on the other hand the course of the dialogue for minutes.
- Summaries and minutes indicate how well the negotiation and translation has performed.

Minutes

Additionally to the summary functionality we worded on another functionality—the generation of *dialogue minutes.* For the minutes, two versions are available in the running system. The first is based on the recognized or translated strings as received from the speech recognizer or given the speech synthesizer. This type of minutes is more or less a word by word recapitulation of



Dialogue summary
Netscape:
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: file:/project/generierung/doc/papers/INLG2000/erg-proto-en.html
What's Related
Internet Lookup New&Cool

VERBMOBIL RESULT PROTOCOL No. 1

Participants: Speaker B, Thompson
Date: 2.2.2000
Time: 5:13 PM to 9:42 PM
Theme: Trip with accommodation and leisure time activities

DIALOGUE RESULTS:

Scheduling:

The meeting is taking place in Hanover. It will take place on the 20. of january 2000 at 11 o'clock in the morning. Speaker B and Thompson are meeting at the train station . This meeting will take place on the 19. of january 2000 at half past 9.

Accommodation:

A good hotel was arranged . The single room costs 80 euro per night . Thompson is taking care of the accommodation .

Entertainment:

Speaker B and Thompson arranged to go out for a meal . Thompson is taking care of the leisure time activities .

Protocol automatically generated at 3.2.2000 09:45:40 h

Figure 4.2: VerbMobil viewed as a summarizer. Message extraction methods are applied to the utterances of the dialogue yielding dialogue act and propositional content (Extraction). These are interpreted in context, forming topic-specific negotiation objects (Interpretation). The most specific accepted suggestions are then processed to produce a summary description (Summary Generation) consisting of formatting directives and German VITs. Depending on target language, the German VITs are eventually sent to the transfer component and, finally, verbalized by the existing generator of VerbMobil–GECO.

the dialogue. The second version tries to utilize a technique used by human interpreters referred to as reduction (see figure 2.1 on page 23), where the negotiation partners tend to be very verbal. A human translator handle this by *reducing* the content of each turn to its central part(s) containing, e. g., intention or moves and propositional content. Our goal with the second version has been to utilize the intentional structure for recognizing the move(s) and attitude(s) and the discourse processor for the computation of the corresponding propositional content for the moves. Outcome of this effort are basic strategies on how the minutes can be generated. Instead of using parts of the thematic structure (see below), the propositional content and/or the dialogue act of each utterance has to be propagated upwards in the intentional structure. The rules for the propagation is implemented in the rules for constructing the intentional structure and obey not only propositional content and dialogue acts but also the structure of the dialogue moves and dialogue games. An example of a rule is the one rephrasing the greeting game, with a standard phrase like "Speaker A and speaker B greeted each other" where "A" and "B" can be replaced by the speaker names in case of successful recognition. The set of rules for the propagation of propositional content upwards in the intentional structure has a quite low coverage. Some functionality, however, for reducing the turns and a first version of the generator has been implemented (see (Alexandersson & Poller, 1998, 2000)).

4.2 A Summarization of Related Work on Summarization

Research within the field of automatic summarization is concerned with, e. g., text summarization (Mani & Maybury, 1999), document extracts (Kupiec, Pedersen, & Chen, 1995), summarization of scientific articles (Teufel & Moens, 2000, 2002) or even meeting summarization (Zechner, 2001b). These approaches are geared towards an open domain or, at least trainable for new domains, and neither provide nor require a detailed model of the domain. Some of them are indeed using machine learning methods, e. g., (Kupiec et al., 1995; Teufel & Moens, 2000) where a set of (annotated) documents are used for training the summarizer. Other work argues that analyzing the text with rhetorical structure theory gives a good basis for the summary, e. g., (Marcu, 1999).

Summarization can be classified into several paradigms. In (Hovy & Marcu, 1998) summarization systems or approaches are discriminated into, e. g., NLP/IE systems which are characterized as follows:

- "**Approach:** try to 'understand' the source - represent content using 'deeper' notation; then manipulate that."

- "**Need:** rules for text analysis and manipulation, at all levels."

- "**Strengths:** higher quality; supports abstracting"

- "**Weaknesses:** speed, still needs to scale up to robust open-domain summarization."

Finally, evaluation is an interesting topic for summarization, e. g., (Mani, 2000-2001; Mani & Maybury, 1999; Mani et al., 1998).

4.2.1 Related Work

We recapitulate some of the more prominent and relevant work on the topic of summarization. The research field of automatic summarization has matured, and a lot of different approaches have been presented in the last decade. For this thesis, we will highlight some of the more relevant approaches. Very few, however, tackle the task of summarizing dialogue, and especially, spoken negotiation dialogue.

Summarization of Scientific Articles

(Teufel & Moens, 2000) describes a method for the summarization of scientific articles. As in VerbMobil, a corpus (of scientific articles) has been annotated which serves as training material. However, the articles have been annotated with rhetorical structures instead of dialogue acts[1] Moreover, to recognize the rhetorical structures a machine-learning method—naïve Bayesian model (Kupiec et al., 1995)—is used. According to the authors, the innovation of their approach is:

> "that it defines principles for content selection specifically for scientific articles, and that it combines sentence extraction with robust discourse analysis."

Output from their system is a list of extracted sentences along with the rhetorical status of each sentence.

[1] Interestingly, the annotation instruction consists of a decision tree similar to that of (Alexandersson et al., 1998).

Summarization of Meetings

At some sites in Europe and the USA, research has turned to investigate meeting room recording and the automatic transcription or summarization thereof. The speechcorder[2], a small device depicted in figure 4.3 containing, e. g., a speech recognizer, is "ICSI's vision" of the future. The vision includes a device which can be put on the table during the meeting and which recognizes and possibly transcribes the meeting. The speechcorder is in itself not a summarizer, but it provides a challenging infrastructure for the automatic generation of meeting minutes and the generation of summaries.

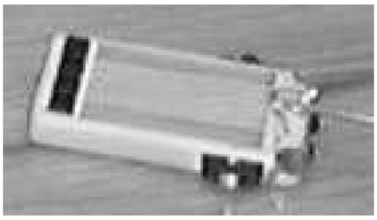


Figure 4.3: The speech recorder device

Other projects concerned with the same topic are, for instance, the *Multi-Modal Meeting Manager - M4* (Moore, 2002), the *Meeting Recorder Project* (Morgan et al., 2002) and the *ISL Meeting Room System* (Waibel et al., 2001). Instead of a small device, the complete meeting room is equipped with microphone arrays, cameras etc. The key tasks are the automatic transcription and summarization of the meeting.

In (Zechner, 2001a, 2002), Zechner describes a challenge, similar to that of the summarization in VerbMobil. There, the summarizing system—DiaSumm—for "automatic summarization of spoken dialogues in unrestricted domains" is described. The approach resembles that of ours in many respects, but the architecture and modeling differs and, since the system works in unrestricted domains, even more shallow techniques are used. Most notably, the output of the system is based on words in the input. The challenges for this system render those of VerbMobil (Zechner, 2001a, p. 2), e. g.,

- "coping with speech disfluencies"
- "coping with speech recognition errors"

[2] See `http://www.icsi.berkeley.edu/Speech/speechcorder.html`

- “identifying coherent topical regions”

The architecture of DiaSumm follows a pipeline approach (see figure 4.4 (Zechner, 2001a, p. 41)); specialized shallow methods are concatenated to produce a summary. Amongst the more interesting techniques described is the following: Zechner takes the confidence scores from the ASR (Waibel et al., 2001) into account to reduce the word error rate in the summary and to enhance the accuracy of the summary. The system does not contain a natural language generator but selects parts of the input to form the summary. The system performance is evaluated against, e. g., “a LEAD baseline, which just includes the first N words of a given topical segment into the summary”, a MMR text summarizer (Carbonell & Goldstein, 1998) and a human gold standard. The metric used is *accuracy* on word level meaning that the system managed to include a certain amount of correct words compared to the gold standard. The system reaches an accuracy between 0.506 and 0.614 on 8 different corpora (Zechner, 2001a, p. 87).

Discussion

Most approaches to summarization are concerned with summarization of texts which can be characterized as monologue discourse. These methods are usually based on identifying interesting parts and then copying parts of the source to form the summary. The selected parts might be further condensed. Work concerned with spoken language and in particular discourse beyond monologue, e. g., (Zechner, 2002) are of more interest to us. However, since the focus is on summarization not restricted to a particular domain, the model of the domain(s) is (are) very shallow or not present at all. Instead other methods for removing irrelevant and redundant information are used.

For knowledge-based methods such as the one we present in this thesis, an elaborated discourse structure and the presense of abstract representation of content is vital for the generation of summaries and minutes. Contrary to open domain summarization as presented in (Zechner, 2002), we can take advantage of this knowledge.

4.3 Natural Language Generation

Our assumption for the summarization functionality in VerbMobil involves a generator that takes structures in the form of propositional content—direx–as input and produces semantic representation—VIT—for sentences. This section is devoted to relevant previous work in generation. We start

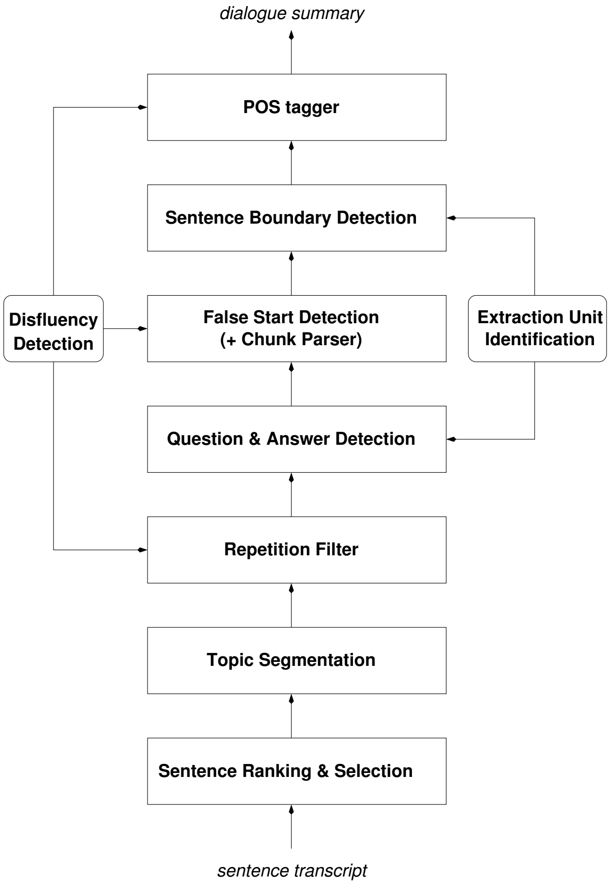

dialogue summary
POS tagger
Sentence Boundary Detection
Disfluency Detection
False Start Detection (+ Chunk Parser)
Extraction Unit Identification
Question & Answer Detection
Repetition Filter
Topic Segmentation
Sentence Ranking & Selection
sentence transcript

Figure 4.4: The pipeline architecture of DiaSumm.

with some terminology and continue to describe some of the more popular and important milestones: Schemata (McKeown, 1985), RST (Mann & Thompson, 1988) and Plan Operators (Moore & Paris, 1993) as well as a well cited system called the ILEX system (Hitzeman et al., 1997), just to argue that the methods are still relevant but all, except RST, make the same assumption: Generation is a top down process *selecting just a relevant part of the knowledge base* in order to fulfill the requirements of the system as a whole. Nevertheless, some of the ideas have been used for our summary generator. Our solution to the generation of summaries resembles the one of (Marcu, 1997) where a *bottom-up* approach to generation is described. There, the generator is faced with a task which, from a certain point of view, resembles that of ours: the complete knowledge base has to be generated. Finally, we classify our summary generator in the light of the results of the RAGS[3] project, e. g., (Cahill et al., 1999).

4.3.1 Some Terminology and Concepts

Within the research field of natural language generation, a big number of concepts and terminology is used. We summarize some of them, later used in this chapter. Most of the material can be found in any text book on generation, e. g., (Reiter & Dale, 2000), and the papers produced by the RAGS project.

Generation

Natural Language Generation (NLG) is the task of mapping some input structure to a possibly formatted output text. There are (Reiter & Dale, 2000):

> "six basic kinds of activity that need to be carried out in going all the way from input data to a final output text"

The six activities are:

- **Content determination**

  Content determination is the process of selecting the relevant part(s) of the knowledge base and possibly mapping this onto a semantic form, possibly annotated with RST relations.

- **Discourse planning**

  Discourse planning (or sentence planning) is the process of mapping conceptual structures onto linguistic ones: this includes generating referring expressions, choosing content words and (abstract) grammatical relationships, and grouping information into clauses and sentences.

- **Sentence aggregation**

  Sentence aggregation is the task of putting entities together into sentences. The effect of sentence aggregation is–if used correctly—enhancement of the fluency and readability of the sentences.

[3] A Reference Architecture for Generation Systems

- **Lexicalization**

  Lexicalization (or lexical choice) is the process of selecting meaning-bearing lexemes. Some researchers argue that this is usually done during syntactic realization (Rambow, Bangalore, & Walker, 2001).

- **Referring expression generation**

  Referring expression generation is the process of determining phrases or expressions to name domain entities. This process is closely related to the process of lexicalization but differs since it consults the discourse history for generation of, e. g., pronouns or demonstratives.

- **Linguistic realization**

  Linguistic realization (or syntactic choice or surface generation) is the process of selecting how meaning-bearing lexemes are combined. Also syntactic choice is closely coupled with lexical choice since the choice of, e. g., a verb affects how the arguments have to be chosen. In some approaches, however, of pronominalization may take place (*Jan likes Jan* vs. *Jan likes himself*).

NLG Architectures

It seems that a consensus about a uniform architecture for natural language generation has emerged (Reiter, 1994; Cahill et al., 1999). One of the questions investigated in the *Reference Architectures of natural Generation Systems* project—RAGS—was whether there is a general reference architecture which all generation systems can be squeezed into (Cahill et al., 1999).[4] In (Reiter, 1994) such an (pipeline) architecture is presented (see figure 4.5) consisting of three modules:

- **Text planner**

  The functionality of this module is usually referred to as "what to say." It comprises the two processing steps, content determination and discourse planning, as described above.

- **Sentence planner**

[4] In (Cahill et al., 2000), it is shown how, in the continuation of the RAGS project, the reference architecture is described in terms of a set of data structure. One of the advantages of their modeling is the separation into a *data model* and a *process model*. Whereas we, in this work, depart from such description, it will be an interesting exercise to characterize our (bottom-up generation) approach with their description.

The sentence planner mostly comprises sentence aggregation, lexicalization, and referring expression generation.

- **Linguistic realizer**

  The linguistic realizer comprises the tasks syntactic, morphological and orthographic processing. For some applications—like the one described here—a formatting functionality may be included.

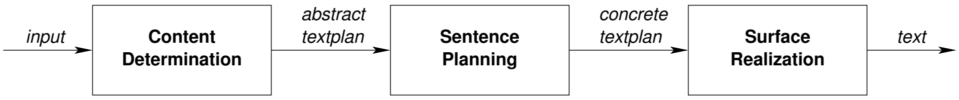


Figure 4.5: Reiter's reference architecture

4.3.2 Related Work

We continue to present some of the more prominent work in the area of automatic natural language generation. We pick out some of the more important milestones which should be relevant for our generation algorithm. The focus of the survey is put on processing strategies.

Schemata

Used within the TEXT generator project (McKeown, 1985), a script-like structure—schemata—provides a explanation strategy for the generation of coherent multi-sentential text achieving a given communicative goal much in the same way humans do. Schemata are built up from *rhetorical predicates* which are used to outline the (structural) relations between propositions in the final text. Following, e. g. (Grimes, 1975), McKeown observed that some combinations of rhetorical predicates occur more often than others. Examples of the way people seemed to produce texts are "analogy with a known concept", "identify the object as being a member of some class", and "evidence supplied for given fact." The rhetorical predicates were combined into four schemata: "Attributive", "Identification", "Constituency", and "Compart and Contrast."

TEXT contains no user model. Rather a "static, casual, and naïve user" was assumed. To generate a text given a communicative goal, the schemata are expanded top-down. Each component of the schema is sequentially accessing the underlying knowledge base to try to fulfill its needs. The fulfillment might result in a recursive call to another schema, or the result of accessing the underlying data base.

Rhetorical Structure Theory—RST

Developed as part of the studies of computer-based text generation, RST (Mann & Thompson, 1988) offers an explanation of the coherence of texts. It emerged through the need for a theory of discourse structure or function that provided enough detail to guide (computer-based) generation of text. As a basis for the studies emerging in RST, a wide range of carefully edited texts was used. It was observed that for every part of the texts there was a reason for its presence on the one hand, and no reason that some parts were missing on the other.

Basis for RST are building blocks at two levels. The first is concerned with *nuclearity* and (coherence) *relations*, whereas the second is concerned with *schemas*. One of the most frequent patterns occuring in texts is that two spans of text are related in some way or another. In RST there is, e. g., an *evidence* relation, where the claim is located in a *nucleus* and the evidence in its *satellite*. A whole range of relations have been identified. In Table 4.1 some of the basic RST relations and their meaning are listed.

Table 4.1: Some RST relations and their meaning

Relation Name	**Nucleus**	**Satellite**
Background	text whose understanding is being facilitated	text for facilitating understanding
Elaboration	basic information	additional information
Preparation	text to be presented	text which prepares the reader to expect and interpret the text to be presented
Contrast	one alternate	the other alternate

Plan Operators

A system has to be able to access previous discourse, or in particular, the system's previous utterances, to be able to participate in, e. g., advisory dialogue. In (Moore & Paris, 1993), it is argued that a discourse model has to include information about the intended effect of previous (system) turns in order to be able to support clarifications of previous misunderstandings. (Moore & Paris, 1993) found that schemata is insufficient for explanatory dialogue, since it does not allow for the representation of the intended effect of the hearer. This is, however, important if the system is supposed to react properly on follow-up questions, since, without knowledge about the intentions of previous contributions, it is problematic to understand what went wrong, or what information was not properly understood by the hearer.

More concretely, Moore and Paris show how a number of rhetorical relations are linked to the intended effect of the hearer. The planner interpreting the plan operators expands these in a top-down fashion (Moore & Paris, 1993, sec. 5.4): Given a communicative goal, e. g., "*Achieve a state where hearer knows about concept c*", the

> "planner identifies all of the potentially applicable operators by searching its library for all operators whose effect field matches the goal."

Since the plan language encodes a decomposition of abstract goals into more concrete ones, the only way to fulfill the given goal is, obviously, top-down expansion. Figure 4.6 shows an example of an operator.

ILEX—Opportunistic Generation

The opportunistic generation approach developed in the ILEX project (Hitzeman et al., 1997; Mellish, O'Donnell, Oberlander, & Knott, 1998; O'Donnell, Knott, Oberlander, & Mellish, 2000) was focusing on automatic text generation. Initially developed for a museum guide scenario, it has been adapted to other domains, like a sales catalogue for computer systems and peripherals. ILEX is a tool for generating "on the fly" from a relational database producing dynamic hypertext comprising text and pictures. The system takes into account not just the discourse history but also factors like the user's level of expertise. The generation process has four stages:

- **Content Selection** In ILEX, the knowledge base has two main sources. In converted form, the first contains information from the original data base. The information is represented in a graph-like structure, called

Plan Operator:

EFFECT: **(PERSUADED ?hearer (DO ?hearer ?act))**
CONSTRAINTS: **(AND (STEP ?act ?goal)**
(GOAL ?hearer ?goal)
(MOST-SPECIFIC ?goal)
(CURRENT-FOCUS ?act)
(SATELLITE))
NUCLEUS: **(FORALL ?goal (MOTIVATION ?act ?goal))**
SATELLITES: **nil**

English Paraphrase:

"To achieve the state in which the hearer is persuaded to do an act,
IF the act is a step in achieving some goal(s) of the hearer,
AND the goal(s) are the most specific along any refinement path
AND the act is the current focus of attention
AND the planner is expanding a satellite branch of the text plan
THEN motivate the act in terms of those *goal(s).*

Figure 4.6: Sample plan operator a la' Moore and Paris. This plan operator is used for persuading the user to do an act.

the *text potential*. Main types in the structure are *i*) representation of objects in entity-nodes, *ii*) facts about the objects in fact-nodes, and *iii*) relations between facts in relation-nodes. The second source captures information collected during interviews with the gallery's curator.

During the content selection stage, decisions about which fact-nodes, additional to the initial fact-nodes, should be verbalized. Different information sources are used to guide the traversal of the graph: The likely interest of a fact for a given user, its importance, and if the user might know this fact already. Some of these constraints can be configured by the user.

- **Content Structuring:** A two-phase discourse structure model is used. In the first, facts are organized into so-called entity-chains. These can be further structured by RST relations into RS trees. In the second phase, all RS trees that can be created from the facts selected are created. For each set of RS trees, the best one is selected and added to one of the entity-chains. This process is repeated until no more RS trees can be added.

- **Sentence Realization** During this stage, the way the facts are to be realized in a collection of sentences is decided. Decisions like tense, mood and surface polarity are taken. An aggregation module makes the text smoother by, e. g. combining groups of facts into a single sentence if possible. Additionally, a NP planning module determines how to realize a noun phrase, i. e. full descriptions, reduced descriptions, or different pronominal expressions.

- **Text Presentation** choosing how to present the sentences to the user.

Bottom-Up Generation

The key insight for introducing *bottom-up generation* (Marcu, 1997) is that there is a difference if *everything that is in the knowledge base* has to be generated, or just a part of it. The latter is almost always the case when a certain communicative goal is given as in, e. g., (Moore & Paris, 1993). Almost all approaches to text planning assumes a tree-like structure or plan and Marcu argues that such *top down* approaches are inadequate if the task of the text planner is to verbalize *all* the knowledge in the knowledge pool. One of the main reasons for this is that schema-step- or plan-operator-driven top-down approaches are unable to predict the amount of the initial knowledge in the selected structure. Thus, it would be possible that, given a construction of a, possibly partial, text encoding a part—and thus not all—of the input structure, the part not used would cause the result to be non-coherent.

Instead of another top-down approach, Marcu presents a bottom-up approach. The basic assumption behind his algorithm is that

> "the knowledge base is represented as a set of semantic units"

and that RST relations

> "hold between pairs of semantic units"

in the knowledge base. Now, the task of generating the full text is solved by a bottom-up strategy where the semantic units are combined in such a way that the final document is coherent with respect to the rhetorical relations between the semantic objects in the knowledge base.

VM-GECO—Constraint Satisfaction

What is constraint satisfaction? In (Tsang, 1993), an informal definition[5] of what a *constraint satisfaction problem* (CSP) is given (Tsang, 1993, page 1):

> "Basically, a CSP is a problem composed of a finite set of **variables**, each of which is associated with a finite **domain**, and a set of **constraints** that restricts the values the variables can simultaneously take. The task is to assign a value to each variable satisfying all the constraints."

Within the VerbMobil project, the multilingual generator—VM-GECO (VerbMobil GEnerator COmponents) (Becker, Finkler, Kilger, & Poller, 1998)—has the task of mapping VITs to surface structure. It consists of two main components. The first—the *microplanner*—has the task of performing lexical and syntactic choice. To do this it uses constraint satisfaction techniques to map the VIT onto a dependency tree. The second—the *syntactic realizer*—is a TAG generator which uses a compiled version of an HPSG grammar. This HPSG grammar is the grammar used by the components within the deep translation track.

More formally, the task of the microplanner is to map a minimal recursive structure or a graph onto a tree structure. This is done by computing a complete covering of the graph with templates thereby computing the result—a dependency tree—on the fly; covering a new part of the graph causes the dependency graph to be extended according to the templates used. The constraints also apply to the construction of the dependency tree, e. g., the filling of obligatory arguments. The overall process consists of two steps. First, the relevant templates for the input VIT are selected. In the second step, a constraint solver—SCREAMER, e. g., (Siskind & McAllester, 1993) and as a more efficient alternative LILIPUT (Becker & Löckelt, 2000)—finds the best solution based on the weighted constraints.

Discussion

As we have seen above, RST is good tool for structuring text especially when it comes to argumentative text, e. g., (Moore & Paris, 1993), comparison of different objects as in the ILEX museum guide, e. g., (Hitzeman et al., 1997; Mellish et al., 1998), or for the system producing larger texts as

[5] The formal definition will require too much space and we therefore satisfy ourselves with the shorter, informal one.

described in (Marcu, 1997). However, the task of the summary generator in VerbMobil is not to produce argumentative or advisory text, but merely to verbalize, from a certain point of view, commitments. The verbalization of these commitments can easily be done using declarative sentences. Sentences in the summary are more or less related to each other. Less, since they are all describing objects that have been agreed upon and which might be distinct objects. Furthermore, if the negotiation objects are so "big" that they cannot be verbalized into one sentence, they therefore have to be split into several sentences and then the need to be connected in some way. Our summaries are comparatively small documents as shown in the Appendix.

To obtain a coherent document, we will need a mechanism for introducing, e. g., pronouns and demonstratives. One simple but functioning approach is to use a simple history list, e. g., (Dale, 1995), which makes it possible to access objects introduced earlier in the generation process.

Finally, but most notably, we are in the same situation as described in (Marcu, 1997): we want to generate the complete knowledge base. Marcu bases his bottom-up generation on the assumption that there are RST relations between objects in the knowledge base. The knowledge base in (Marcu, 1997) seems to consist of bigger atomic structures than ours. A related issue in the sense that the whole input structure has to be processed, is described in (Becker et al., 1998). There, the generation task is, formally, to map the entire graph to a dependency tree, or in other words: to solve a complete covering problem using templates, and at the same time produce the output.

4.4 The Summary Generator - SuGe

This section is devoted to the the implementation of the multilingual summary generator. Our implementation platform is the plan processor described in section 3.8.1. We encode the actual generation algorithm as well as the linguistic knowledge using plan operators. The linguistic knowledge consists of both compiled information from the semantic database and handcrafted knowledge which has been created for the generation of summaries. In the next section (4.4.1) we present the architecture of the final system. The sections 4.4.2 and 4.4.3 are concerned with requirements, concepts and the actual implementation.

4.4.1 Requirements and a Solution

For the summary generator we have collected a number of requirements which should serve as a guideline for the conceptualization and implementation. The list is a mixture of theoretical considerations and practical ancillary conditions:

- **The generation algorithm** As we have seen above, the most reasonable way to generate a summary is to use a bottom-up strategy. The main reason is that we want to generate everything selected from the discourse memory.

- **Multilingual summaries** A basic requirement is to be able to generate multilingual documents. VerbMobil is a translation system and thus contains several translation facilities. One of these is the transfer module which follows a semantic transfer approach by translating VITs to and from any language pair implemented in the system.

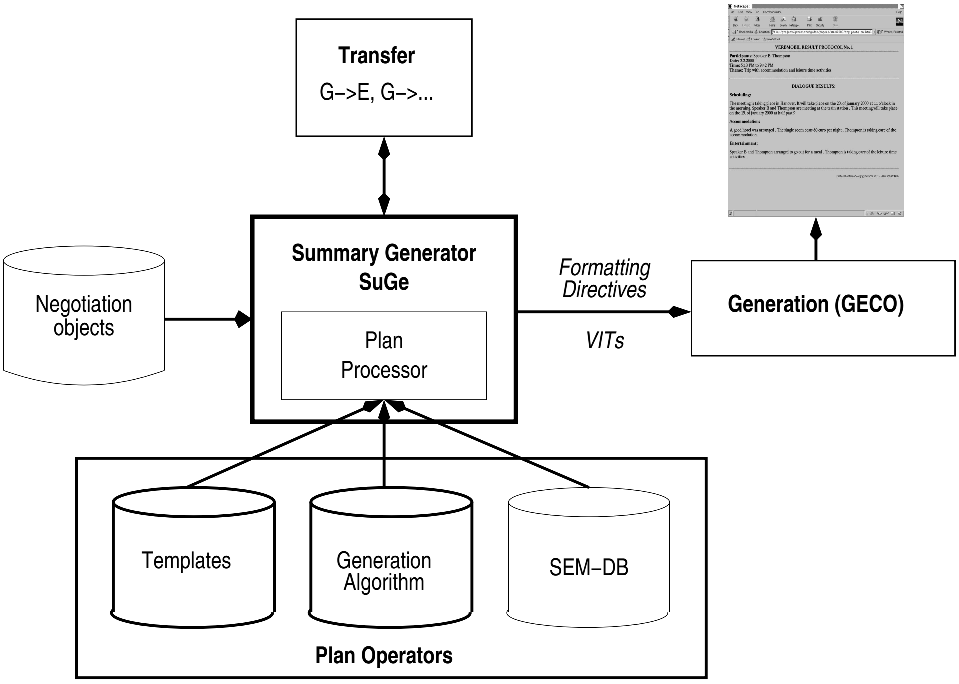
- **Existing software** The dialogue module contains a plan processor which was developed to a general purpose tool. As shown in section 3.8.1, the plan processor can be used for implementing plan operators in the style of (Moore & Paris, 1993) but also as a Prolog interpreter.

- **Robustness** We would like the generator to be robust. An extension of the domain model should not cause the generator to fail *and* we would like the extensions to show up in some way in the final document.

- **Appearance** The finally document should be available as ASCII text, HTML and LaTeX. The latter provides the basis for a number of other formats, e. g., postscript and PDF.

- **Stylistics** The summary should function as a reminder for the dialogue participants. Thus, the content of the summary should be polite but does not need to contain additional information, like, who proposed what.

In trying to obey as many items of the above-mentioned requirement as possible, we use the approach depicted in figure 4.7. First, in a "what-to-say" step, the *most recent specific accepted negotiation objects* are selected from the dialogue memory. Then, the negotiation objects are mapped into a sequence of intermediate representations corresponding to sentences. The generation algorithm and the knowledge sources are implemented using our plan processor. Then, the sentence descriptions are, in the third step, processed to obtain a smoother and more coherent text by introducing anaphora and demonstratives. In case the target language is not German, the VITs



VERBMOBIL RESULT PROTOCOL No. 1

Participants: Speaker B, Thompson
Date: 2.2.2000
Time: 5:13 PM to 9:42 PM
Theme: Trip with accommodation and leisure time activities

DIALOGUE RESULTS:

Scheduling:

The meeting is taking place in Hanover. It will take place on the 20. of january 2000 at 11 o'clock in the morning. Speaker B and Thompson are meeting at the train station . This meeting will take place on the 19. of january 2000 at half past 9.

Accommodation:

A good hotel was arranged . The single room costs 80 euro per night . Thompson is taking care of the accommodation .

Entertainment:

Speaker B and Thompson arranged to go out for a meal . Thompson is taking care of the leisure time activities .

Protocol automatically generated at 3.2.2000 09:45:40 h

Figure 4.7: Conceptual Architecture of the Summary Generator—SuGe—in VerbMobil. The new parts are marked with thicker lines: The actual summary generator and the templates.

are sent to the transfer component for translation. Finally, the document specification is generated and formatted by the VM-GECO generator.

In the following sections we will present the various steps in more detail. The next section (3.8.1) describes the generation algorithm used for mapping the negotiation objects onto VITs. The actual implementation is discussed in section 4.4.2 followed by a section describing some observations made during the development of the summary generator.

4.4.2 Designing the Generation Algorithm

As this part of the project started, we took the approach to generation using plan operators. Inspired by, e. g., (Moore & Paris, 1993), we used our plan processor and wrote operators for each concept of the domain model taking into account what we thought was *the only way(s) to generate an instance of this kind*. The operators were expanded using the standard Prolog top-down and left-to-right algorithm. Very soon, in fact, as soon as the domain model started to really evolve, we experienced the following:

- Every time the domain model was *extended*, the new information never showed up in the summary. The first obvious reason for this was that we had not coded the new roles in the plan operators. The second was that, since our implementation did not force an operator or the set of operators for a particular concept to fail unless all information had been consumed, the summary was generated—but with the same content as before the extension of the domain model.

- Almost every time the domain model was *changed*, i. e., roles for a certain concept were removed or moved, the change caused the plan operators concerned to fail and we ended up with a small or an empty summary not reflecting the accepted negotiation objects in the dialogue memory. Consequently, we were forced to rewrite some of the plan operators to account for the new design. This was *very* time consuming.

Therefore we started to look for alternative robust and flexible methods for generation. Our second and present approach is based on two observations:

- To guarantee that really everything in the knowledge base is generated, we have to implement the algorithm in such a way that our rules or templates cover *all* of the structure. Becker et al. (Becker et al., 1998) show one way of doing this.

- In the generation literature—as mentioned above—the bottom-up generation is more appropriate for such cases, e. g., (Marcu, 1997).

Some Observations and Terminology

In order to be precise we recapitulate and describe some concepts important for the presentation below.

Function Words Function words are closed-class words. They have no or less semantic content and serve more a grammatical purpose. There are only about 300 in English, such as:

Function Words	**examples**
Prepositions	of, at, in, without, between
Pronouns	he, they, anybody, it, one
Determiners	the, a, that, my, more, much, either, neither
Conjunctions	and, that, when, while, although, or
Modal verbs	can, must, will, should, ought, need, used
Auxiliary verbs	be (is, am, are), have, got, do
Particles	no, not, nor, as

Content words Content words are open class words (new words are being added in every language). Most of them function as carriers of semantic content. Some examples are:

Content Words	**examples**
Nouns	John, room, answer, Selby
Adjectives	happy, new, large, grey
Full verbs	search, grow, hold, have
Adverbs	really, completely, very, also, enough
Numerals	one, thousand, first
Interjections	eh, ugh, phew, well
Yes/No answers	yes, no (as answers)

The VerbMobil **semantic database—SemDB** Organized around semantic classes (Bos, Schiehlen, & Egg., 1996) (see below), the VerbMobil semantic database (Heinecke & Worm, 1996) is used to classify lemmata. The information is used by a number of modules in VerbMobil concerned with semantic processing.

Abstract semantic classes (Kasper, Bos, Schiehlen, & Thielen, 1999) describes the abstract semantic classes used in VerbMobil. For the summary generation, we are particularly interested in function words, e. g., prepositions, verbs and their frames and a subset of other content words.

Verb frames In the semantic database, besides belonging to a class, e. g., transitive verb—tv, the verb entries have information about its frame. Important for the summary generator is the sortal restrictions on the arguments. For the processing within the deep translation track all arguments are facultative due to the fragmentary characteristics of the processing as well as the fragmentary characteristics of spontaneous speech. In our usage of the verbs, we view the arguments as non-facultative, i. e., all arguments of a verb has to be filled. Below, we will refer to a verbs class as, e. g., the *pcv13 vereinbaren*, which is to be read: The verb “vereinbaren” is of the class pcv13, i. e., propositional complement verb, and has two arguments, namely arg1 and arg3. Table 4.2 shows the translation of the semantic verb classes and their suffix together with some German examples.[6] The *treffen* entry of the semantic database contain, for example, PredName (treffen), SemClass (v13) SynthFrame (arg1:subj=agent/nom,arg3:obj=theme/acc), Sort (meeting_sit) and ArgSorts (human,human).

Other content words Other entries of interest to us are, for instance *nominals*, e. g., dofw, mofy, yofc, ctime and pron, *isect*, e. g., ta (heute) and *mood*, e. g., decl.

Table 4.2: The verbs of the VerbMobil semantic database and their corresponding suffix.

Verb Class	Suffix	Example (G)
verbal	**v**	stat
v_arg2	**v2**	rechthaben
v_arg3	**v3**	kommen
v_arg12	**v12**	schliessen_unspec
v_arg13	**v13**	eintragen
v_arg23	**v23**	passen_unspec
v_arg123	**v123**	aussuchen
pcv_arg13	**pcv13**	notieren
pcv_arg23	**pcv23**	finden
pcv_arg123	**pcv123**	erinnern_an
pcv_arg3	**pcv3**	klingen
modal	**mv**	koennen
underspecified_verbs	**unspec**	gehen_unspec

In the VIT representation we have access to various predicates and constructions. Most of them have a more or less direct realization on the surface, but our part of the summary generator as a whole has no control over what

[6] The class negated_modal has been omitted, since there are no German verb of this kind. An English example is “cannot”

is really serialized. Therefore, in what follows, we will use the expression *discourse referents* to denote content words. Specializations of discourse referents are *nominal objects*, *prepositional objects* and *sentential objects*. These concepts roughly correspond to NPs PPs and sentences respectively (see above).

Having set some terminology, we turn to two observations made in the introduction of this section.

- Distinct parts of the DIREX structures can be converted to a distinct discourse referent—abstract or concrete—naming, e. g., events or locations. Clearly, the size and structure of these parts differ.
- The target discourse referent depends on the context in which they occur, or, in other words: there are certain conditions that allow or prevent an object to be converted to a certain referent or not. These will be referred to as *conditions*.
- There are certain rules for *how* a part is converted. These rules will be called *mappings*.

In the next section we will present our algorithm for checking the conditions before applying the mappings.

4.4.3 Implementing the Generation Algorithm

Our bottom-up generation algorithm can be viewed as building a three-dimensional Lego construction using pieces with different patterns, i. e., some pieces can be put together and some cannot. The selected structures from the dialogue memory form the foundation of the construction. The conditions function as Lego pieces that have certain texture on the bottom and the top. The mappings also have texture on the bottom. On the top, however, there might be texture but also a part that functions as a roof of the puzzle. It is impossible to put other pieces on the roof. Using this metaphor, generation consists of adding Lego pieces—conditions and mappings—to the construction—input structures—in such a way that the construction is completely covered. If the top pieces are sentential objects it was possible to completely generate the input structure. The order in which we “harvest” the top pieces depends on the order in which we build them.

The Implementation Platform

We have used our plan processor to implement the generation algorithm. For this application we utilize the ability of the plan processor to mimic the prolog interpreter. In what follows we will thus use the terms *predicate* instead of "plan operator", *head* instead of "goal" and *body* instead of "subgoals" interchangeably.

The present implementation of the algorithm consists of three steps:

1. During this step, we search for a complete mapping, projecting the input structure onto a sequence of sentential objects. The details are revealed in the next three sub-sections.

2. The sequence of sentential objects are post-processed to obtain a coherent sequence. Coherence is achieved by deciding whether an nominal object should be definite or not in case it has not been decided yet. Other decisions involves the introduction of demonstratives and pronominalization.

3. The final step is to translate the discourse referents to VITs.

The Search Algorithm

The main predicate of our search algorithm consists of one recursive plan operator[7]—`gen-concept/2`—as depicted in figure 4.8. It is invoked with an instance of some top-level concept we call the *root*. It starts by resolving the goal `gen-daughters/2` which returns a triple consisting of sentential, nominal and prepositional objects. The next step is to resolve the goal `make-sententials/4` which is passed the two latter output argument of the former goal together with the input structure to `gen-concept/2`. The output position is bound to a list of sentential objects. Finally, all sentential objects are appended and returned. If `gen-daughters/2` or `make-sententials/4` fail, the algorithm backtracks using chronological backtracking. The search is guided by adding numerical values to the predicates. This navigates the search algorithm to maximize the result based on *local* preferences. Examples of such preferences are the number of constituents in one sentential object, or to the preference of certain verbs while converting certain concepts.

As the search is completed, the sentential objects are post-processed in order to obtain a coherent sequence of sentences. Decisions, such as, whether a nominal object should be definite or not (unless this has not been decided yet) are made. Other decisions are the introduction of demonstratives and anaphorical expression. These decisions are made based on the content of the objects, the objects in the input they are based on and the history list.

[7] "Some" details have been omitted in favour of clarity.

```
(defplan gen-concept
  :goal (gen-concept ?prop-cont ?sent-objects)
  :subgoals (:seq
              (gen-daughters ?prop-cont
                             (?s-objs ?n-objs ?a-objs))
              (make-sententials ?prop-cont ?n-objs ?a-objs ?_s-objs)
              (s-append ?_s-objs ?s-objs ?sent-objects)))
```

Figure 4.8: The main predicate of the generation algorithm—`gen-concept`.

```
PC-APPOINTMENT (P21**+0)
  HAS_DURATION --> PC-DURATION (P6******)
                     TEMPEX=tempex(ge_2753_735,[for:dur(1:1:2,days)])
  HAS_DATE --> PC-DATE (P21**)
                 TEMPEX=tempex(ge_2753_745, from:[year:1997,
                                                  month:july,
                                                  dom:5,
                                                  pow:weekend])
  HAS_PARTICIPANTS --> (#<PC-VM_SPEAKER SPEAKER1>
                        #<PC-VM_SPEAKER SPEAKER2>)
```

Figure 4.9: A sample input structure for the summary generator. The root (`pc-appointment`) is a meeting with three filled roles: `has_duration, has_date and has_participants`. A possible verbalization is "*Speaker1 and speaker2 meet on the fifth of July. The meeting lasts 1.5 days.*"

Consider the sample structure in figure 4.9. The structure represents a meeting with two participants, a certain duration and date. A possible realization of this object is "*Speaker1 and speaker2 meet on the fifth of July. The meeting lasts 1.5 days*", i. e., we have mapped the structure onto two sentences. In this case, "*the meeting*" in the second sentence is definite since it realizes the same object—`pc-appointment (P21***+0)`—as the verb "*meet*" in the first sentence.

Conditions and Specifications

Conditions allow the mappings to actually convert parts of the input structure and already created discourse referents to new discourse referents. The conditions are all declarative facts describing information about what constructions are allowed. A part of the conditions are *specifications* which are recipes for how a certain mapping should be done. Figure 4.10 (see page 159)depicts some of the major conditions used for generating a certain set of characteristic appointment objects. The first—`class-verb-relation`—states that

- we are allowed to use either the V3 verb *stattfinden* (*happen, to occur* or *to take place*) or pcv13 *vereinbaren* (*agree upon*) together with the values of some combination of the roles `has_location`, `has_dest_location`, `has_meeting` and `has_date`. The combinations are expressed with the operators `:or, :xor, :sor, ;seq` and `:optional` (see section 3.8.1).

- Alternatively, we are allowed to use the v13 verb *treffen* (*meet*) together with value of the one of the roles `has_location` or `has_dest_location`. Optional the (pcv3) verb *dauern* (*to last*).

The second condition—`class-verb-role`—states that an object of type `pc-appointment`[8] realized with the verb `treffen` can additionally be generated with the speakers of the dialogue in the subject position of the verb and the reflexive pronoun in the object position. This is necessary, since the DIREX does not contain a participants role, and we will have to get the participants from the situative context. The third—`adj-mod-fact`—states that it is possible for the value of the `has_date` role to be adjuncted to the verb via the `temp_loc` relation. Finally, the fourth fact—`verb-frame`—defines

[8] The prefix "pc-" stands for propositional content and is used to refer to instances of our domain model.

the valence role of a verb, i. e., it states what arguments a verb, in this case `treffen`, has together with the conditions on the arguments. These facts are automatically derived from the VerbMobil semantic database.

Worth noting is that the use of the logical operators `:xor` etc. allows for a very compact notation. The `class-verb-relation-appointment` operator depicted in figure 4.10 expresses 58 different combinations.

Mappings

Given applicable conditions, a mapping converts input structures and discourse referents to discourse referents. For a certain mapping to be applicable, certain conditions must be met, i. e., to create a nominal object, a specification has to be present. Figure 4.11 shows two examples of mappings. The first mapping—`make-s-object`—is used to make a sentential object by first consuming the roles—`set-roles/3`—and, in case there are any roles left over, additionally trying to attach the rest of the roles as adjuncts. The two latter is an example of a specification–mapping pair. In this case, the `class-fact-pc-city` states that a `pc-city` should always be mapped onto an indefinite nominal object using the value of the role `has_name`. Cities are examples of objects which in our domain always are indefinite.

Some concepts of the domain model can be converted to nominal objects using `class-fact` as well as sentential objects using `gen-concept`. An example is the concept `pc-room` where an instance of the concept can be empty or contain just the role `has_size`, but also additional roles, like `price`. Containing several roles usually means that the object is realized as a sentential object.

Additionally, we make use of the following knowledge and constructions:

- Competing plan operators processing a certain concept are automatically sorted according to the specificity of its concept.

  ```
  (deffact pc-location
      :prob (class-prob 'pc-location)
      :constraints (typep !pc-location 'pc-location)
      :goal (class-fact
              ?pc-location
              (named has_name :def-or-udef def)))
  ```

  The code `(class-prob 'pc-location)` computes a number which is used to rank the operator. For class-facts, a more special realization is preferred.

```
(deffact class-verb-relation-appointment
  :constraints (typep !pc-appointment ’pc-appointment)
  :goal (class-verb-relation
         ?pc-appointment
         ((:xor
           ((:xor stattfinden (vereinbaren :tense past))
            (:optional :sor (:xor has_location has_dest_location)
                       has_meeting has_date))
           (treffen
            (:sor has_date (:xor has_location has_dest_location)))
           )
          :optional
          (dauern
           (:sor has_duration)))))

(deffact class-verb-role
    :goal (class-verb-role appointment
                           treffen
                           ((subj speakers)
                            (obj refl-pron))))

(deffact adjunct-modification
    :goal (adj-mod-fact treffen ?pc-type
                   adj (has_date) (pp temp_loc)))

(deffact verb-frame
    :goal (verb-frame
            treffen
            ((subj nom human)
             (obj acc human))))
```

Figure 4.10: Conditions for relating the generated values of roles of an appointment to a verb.

```
(defplan make-sentential-object
   :goal (make-s-object ?sent-spec ?sent-obj)
   :subgoal (:seq (set-roles ?sent-spec ?_sent-obj ?left-overs)
                  (set-adjuncts ?_sent-obj ?left-overs ?sent-obj)))

;; Specification:
(deffact class-fact-pc-city
    :constraints (typep !pc-city 'pc-city)
    :goal (class-fact
           ((?pc-city has_name :def-or-udef udef)))
;; Mapping:
(defplan make-n-object
    :goal (make-n-object ?rel ?n-obj)
    :subgoals (:seq (class-fact (?rel . ?specs)))
    :actions (== ?n-obj (make-n-object !rel !specs)))
```

Figure 4.11: Some examples of mappings.

- `rel->n` recipe for a relation name to a noun

  ```
  (deffact rel->n
    :goal (rel->n HAS_THEATER_PLAY theaterstueck))
  ```

- `class->n` recipe for a class to a noun

  ```
  (deffact class->n
      :goal (class->n pc-appointment treffen))
  ```

- `match` The predicate match allows for checking the "neighbourhood" of the object. It checks a pattern by walking up and down starting at a given concept. Below are two examples. In the first, the move can be mapped to a nominal object representing *eine Bahnfahrt—a train journey* if it contains the role `has_transportation` with an object of type `pc-rail`. The rail object is not allowed to contain a `has_name` role. In the second example, "`(up 2)`" climbs two steps, i. e., by way of the role pointing at the move to the concept containing that role. The move can thus be mapped to *ein Hinflug—a outward flight*—if the move is pointed to by a `has_move_there` role, and has the role `has_transportation` containing a plane without name.

```
(defplan move->n
    :constraints (typep !move 'pc-move)
    :goal (object->n ?move bahnfahrt :def-or-udef indef)
    :subgoals ((match ?move
                 (has_transportation pc-rail (not has_name)))))
(defplan move->n
    :goal (object->n ?move hinflug :def-or-udef indef)
    :subgoals ((match ?move
                 ((up 2) has_move_there ?move)
                 (has_transportation pc-plane (not has_name)))))
```

Generating time expressions

Our generator makes use of a generation component developed during the first phase of VerbMobil. This component can translate TEMPEX-es into nominal or prepositional objects and is therefore called the *tempex generator*. The tempex generator is implemented using a bottom-up LALR1 parser. The parser composes the target structure in a compositional way much like semantics are compositionally composed in, e. g., an HPSG grammar.

Representation of Sentential Objects

The system represents sentential objects as a list containing the verb and its argument frame. The argument frame is itself a list of phrasal constituents which, in the present version, contains only nominal and prepositional objects. Each constituent is a list with the following features:

- **grammatical role** Values: subject, object and adjunct.
- **case** Values: nom gen dat acc.
- **sort** The most general semantic sort of the constituent (see page 214 for the complete inventory of sorts in VerbMobil).
- **realized entities** The discourse entities realized by this phrase.
- **anaphorized entities** Identifiers of earlier phrases that might have been anaphorized at this point.

Interfacing GECO

Our interface to the GECO generator consists of a tuple as follows:

1. **location** A string representing the place of the negotiation
2. **date** A string on the form `(dd mm yyyy)` representing the date of the negotiation
3. **theme** A VIT mirroring the theme of the negotiation. An example is *Appointment Schedule with trip and accommodation.*
4. **begin-time** A string of the form `(hh mm)` representing the begin time of the negotiation
5. **end-time** A string of the form `(hh mm)` representing the end time of the negotiation
6. **participants** An list of pairs on the form `((<tag> . VIT) ...)` possibly containing a VIT corresponding with the names including titles of the participants. In case the names are not known, the speakers are called `"Speaker A"` where `A` is the name of the channel used by the speaker.
7. **topic-results** A quadruple of lists with sentence description sorted according to the four topics *scheduling*, *travelling*, *accommodation* and *entertainment.* Each of the lists has the form `(<topic-tag> VIT*)`

Discussion

From a purely technical point of view, our algorithm for mapping the selected parts of the dialogue memory to sequences of VITs is related to constraint satisfaction. The informal definition of CSP (see page 147) states that a set of variables where each variable is associated with a finite domain and a set of constraints. In our case, the mappings can be seen as these variables. The specifications, conditions and, most notably, the selected structure from the discourse memory stand for the constraints.

Some advantages with our approach are:

- **Robustness** The algorithm is robust against changes and extensions of the domain model. Since we have separated the search from realization, the knowledge base—conditions and mappings—has to be modified and possibly extended.
- **Flexibility** Structural changes of the domain model do not affect the result of the generation to the same extent as before. The usual effect of moving a role within the model is that the objects associated with that role appear somewhere else in the output structures.

- **Expressibility** Since our specifications of what can be generated is constrained only by the linguistic capabilities of the content words—especially the verbs—we are almost always guaranteed success in generating a semantic representation of the summary.

Compared to the partition of the generation process as described by (Reiter & Dale, 2000), our summary generator omits the *Sentence aggregation* step. Worth mentioning is that the *referring expression generation* is carried out during the summary generation step, and not later in the processing pipeline. Compared to the three-step reference architecture posed by Reiter (see figure 4.5) SuGe performes the two first steps whereas the surface realization is taken care of by GECO. Finally, a reimplementation of the ideas presented here must allow for a bidirectional protocol between SuGe and GECO. Succeeding in generating a semantic representation does not necessarily imply success in the realizing step.

4.5 Evaluation

Evaluation is hard and challenging. There are many reasons for this (e. g., (Mani & Maybury, 1999)) and we will add yet another one at the end of this section. For negotiation dialogues, there is still a hope that the agreed-upon negotiation objects can be detected by a human reading the transcriptions and we can thus base our evaluation on a fairly well-grounded "gold standard." For the summary machine (see figure 4.1) in general, however, the challenge remains. In our case, things are further complicated by the nature of a speech-to-speech translation scenario. There are a lot of system errors that can lead to a possible partial breakdown of the dialogue and subsequent repair dialogues posing additional challenges to the components trying to understand the dialogue. When analyzing multilingual off-line data using the recognized and translated utterances as a basis, it is sometimes almost impossible to judge what has actual been agreed upon. Consider the excerpt from one of our German–English evaluation dialogues in figure 4.12, where for both participating speakers and an observer it is difficult to grasp what is going on in the dialogue. In (82) the translation works pretty well, but in (83) the intention has been—despite perfect recognition—erroneously recognized and the system replaced a request with a suggest. For the final utterance of the excerpt (84) the processing finally completely breaks down and the translation has nothing to do with the spoken input.

In (Reithinger et al., 2000) we used four transcribed German–English dialogues which had been mediated by the system as the basis of evaluation.

(82) okay then, that sounds fine. midday at twelve?
and sounds fine meet at twelve then

ja gut dann lass uns doch um zwölf Uhr treffen
"well ok let us meet at twelve"

(83) aber an welchem Tag?
aber an welchem Tag?

"but that day suits me as well"
"but on what day?"

(84) which day did you want to leave? at midday.
weeks tight did you want to me at night out

wollten Sie für mich erkennen
"Would you recognize for me"

Figure 4.12: Excerpt from one of the German–English evaluation dialogues. Each block shows the spoken utterance (first row), recognized chain (second row), *system translation* (third row) and *translation* (fourth row—82 and 84 is a translation of the system translation whereas 83 is a translation of the spoken utterance).

As "gold standard" we use the following procedure. For each of the transcribed dialogues, a human marked the agreed on features, maximally 75, e. g., location, date for a meeting, speakers name and title, booking agent.[9] The dialogues were then run through the system, and summary was generated. Finally, we compared the features in the summary using standard classifications as described in (Mani & Maybury, 1999):[10]

- **True Positive—TP** The Feature approximately corresponds to the gold standard. This means that the feature is either (1) a 100% match, (2) not specified enough or (3) too specific[11].

- **False Positive—FP** A feature was not part of the result of the negotiation, but was included in the summary.

- **True Negative—TN** A feature was not part of the result of the negotiation, and not included in the summary.

[9] Each dialogue only contain a subset of these features.

[10] In (Reithinger et al., 2000) we referred to these classifications as **Corr**, **Miss**, **TN** and **False**.

[11] Example of (2) is when the correct date included a time, which was not captured. Example of (3) is when a date with time was annotated but the feature contained just a date.

- **False Negative—FN** A feature received an incorrect value or was not included in the summary despite being part of the result of the dialogue.

These definitions can be visualized using the *Category Task Contingency Table* (see table 4.3) where X, Y are two distinct features, and ϕ represents no feature.

Table 4.3: The Category Task Contingency Table, visualizing **TP, FP, TN** and **FN**. The two columns stipulate the content of one feature of the dialogue. Either a feature (X) is present or not (ϕ). The rows constitute two distinct features X and Y, or no feature (ϕ).

	Gold Standard	
Summary	X	ϕ
X	**TP**	**FP**
Y	**FN**	**FP**
ϕ	**FN**	**TN**

For the evaluation we use the following (standard) metrics as defined in, e. g., (Mani et al., 1998):

- **Precision** The percentage of the correctly recognized features related to the total number of features in the final summary.

$$Precision = TP/(TP + FP) \tag{4.1}$$

- **Recall** The percentage of the correctly recognized features related to the number of correct features in the gold standard.

$$Recall = TP/(TP + FN) \tag{4.2}$$

- **Fallout** The percentage of erroneously recognized features related to the total number of features in the final summary.

$$Fallout = FP/(FP + TN) \tag{4.3}$$

- **F-score** The harmonic mean of precision and recall.

$$Fscore = Precision * Recall/(Precision + Recall) \tag{4.4}$$

The result of our evaluation is shown in figure 4.13. In (Reithinger et al., 2000) we concluded that our approach tries to be on the safe side; the summary contains only those features that the system thinks both partners agreed on. The main reasons for not getting better numbers is due to the limited recognition of dialogue acts (70% recall) and errors in the content extraction.

Dialogue	1	2	3	4	aver
Turns	33	33	31	32	32.25
TP	6	13	9	11	9.75
FP	6	3	5	4	4.5
FP	3	3	3	0	2.25
TN	32	28	30	32	30.5
Precision	0.67	0.81	0.75	1.0	0.81
Recall	0.50	0.81	0.64	0.73	0.67
Fallout	0.09	0.10	0.09	0.00	0.07
F-score	0.57	0.81	0.69	0.84	0.73

Figure 4.13: Evaluation Results for four bilingual German–English dialogues assuming perfect speech recognition.

For the work described here we have re-evaluated the summary functionality. Basically the same procedure has been used, but instead of assuming perfect recognition we introduced yet another source of error and used transcribed dialogues as recognized by our speech recognizers. Also the human was given these transcripts as a source for setting the gold standard. Altogether, we have evaluated 30 dialogues—20 monolingual (10 German–German and 10 English–English) as well as 10 bilingual German–English—using the same procedure as described above. The result is shown in figure 4.14.

As can be seen, our F-score is fairly good throughout all dialogues: 0.49 for the mono-lingual dialogues and 0.59 for the bi-lingual ones. For the German–German dialogues we have a better recall than for the English–English—0.45 compared to 0.38. However, for the English–English dialogues we have a better precision than for the German–German—0.80 compared to 0.58. Still, the fallout remains low, especially for the multi-lingual dialogues: 0.02.

The main reason for better results for the bilingual dialogues is that they render a simpler dialogue structure than the mono-lingual dialogues. The effects of non-perfect translation by the running VerbMobil system force the participants to use simpler and robuster strategies for the negotiation.

	Monolingual (German–German)										
D.Nr.	1	2	3	4	5	6	7	8	9	10	avg
Turns	40	43	45	52	45	43	33	25	52	52	43.00
TP	5	6	7	3	15	5	8	11	6	10	7.60
FP	7	5	3	4	6	5	0	6	10	12	5.80
TN	81	83	82	86	76	85	89	82	79	75	81.80
FN	11	10	12	11	7	9	7	5	9	7	8.80
Prec.	0.42	0.55	0.70	0.43	0.71	0.50	1.00	0.65	0.37	0.45	0.58
Rec.	0.31	0.37	0.37	0.21	0.68	0.36	0.53	0.69	0.40	0.59	0.45
F-out	0.08	0.06	0.04	0.04	0.07	0.06	0.00	0.07	0.11	0.14	0.07
F-sc	0.36	0.44	0.48	0.29	0.70	0.42	0.70	0.67	0.39	0.51	0.49

	Monolingual (English–English)										
D. Nr	1	2	3	4	5	6	7	8	9	10	aver
Turns	72	76	51	42	44	81	62	89	66	61	64.40
TP	10	4	5	5	7	7	6	7	8	6	6.50
FP	0	1	1	1	0	3	0	0	15	7	2.80
TN	89	82	82	87	85	82	86	86	74	82	83.50
FN	5	17	16	11	12	12	12	11	7	9	11.20
Prec.	1.00	0.80	0.83	0.83	1.00	0.70	1.00	1.00	0.35	0.46	0.80
Rec.	0.67	0.19	0.24	0.31	0.37	0.37	0.33	0.39	0.53	0.40	0.38
F-out	0.00	0.01	0.01	0.01	0.00	0.04	0.00	0.00	0.17	0.08	0.03
F-sc.	0.80	0.31	0.37	0.45	0.54	0.48	0.50	0.56	0.42	0.43	0.49

	Multilingual (German–English)										
D. Nr.	1	2	3	4	5	6	7	8	9	10	avg
Turns	13	11	42	14	24	22	30	13	14	14	19.70
TP	8	13	11	3	6	3	3	5	6	15	7.30
FP	3	0	0	4	8	4	0	0	2	0	2.10
TN	88	85	87	96	85	87	91	95	83	81	87.80
FN	5	6	6	1	5	10	10	4	13	8	6.80
Prec.	0.73	1.00	1.00	0.43	0.43	0.43	1.00	1.00	0.75	1.00	0.78
Rec.	0.62	0.68	0.65	0.75	0.55	0.23	0.23	0.56	0.32	0.65	0.52
F-out	0.03	0.00	0.00	0.04	0.09	0.04	0.00	0.00	0.02	0.00	0.02
F-sc.	0.67	0.81	0.79	0.55	0.48	0.30	0.37	0.71	0.44	0.79	0.59

Figure 4.14: Evaluation results for 30 (10 English, 10 German and 10 German–English) dialogues using the output from our speech recognizers and segmentation by the prosody module.

Manually transcribed						
Dialogue	1	2	3	4	5	avg
Turns	37	29	45	48	36	39.00
TP	9	9	15	14	15	12.40
FP	9	0	12	19	10	10.00
TN	77	88	70	61	70	73.20
FN	9	7	7	10	9	8.40
Precision	0.50	1.00	0.56	0.42	0.60	0.62
Recall	0.50	0.56	0.68	0.58	0.62	0.59
Fallout	0.10	0.00	0.15	0.24	0.12	0.12
f-score	0.50	0.72	0.61	0.49	0.61	0.59

Speech recognized						
Dialogue	1	2	3	4	5	avg
Turns	43	33	50	52	52	46.00
TP	5	8	9	6	10	7.60
FP	5	0	12	10	12	7.80
TN	85	89	71	79	75	79.80
FN	9	7	12	9	7	8.80
Precision	0.50	1.00	0.43	0.37	0.45	0.55
Recall	0.36	0.53	0.43	0.40	0.59	0.46
Fallout	0.06	0.00	0.14	0.11	0.14	0.09
f-score	0.42	0.70	0.43	0.39	0.51	0.49

Figure 4.15: Evaluation of five German–German dialogues, manually transcribed and processed by speech recognition.

For the mono-lingual dialogues the participants are speaking spontaneously using whatever expression and think-aloud language they like.

The Effect of Speech Recognition

To see what effect speech recognition has on our summarizer, we have randomly picked 5 German–German dialogues and summarized the manual transcripts and the dialogue processed by one of our speech recognizers. The evaluation is depicted in figure 4.15.

As can be seen, we lose 7% precision and 13% recall and consequently the F-score goes down 10%. This is mainly because the true positive goes down almost 39%! Still, the performance of the summarizer is almost equal for dialogue 2, whereas a big loss in precision and recall can be seen for the rest of the dialogues. Finally, the fallout remains fairly stable (−3%) and low, indicating that we still do not select too many features erroneously.

4.5.1 Discussion

Even though it is not straight forward and possibly even a debatable process, we compare our results with those of (Zechner, 2001b) keeping in mind that the approach and indeed the data processed by DiaSumm and VerbMobil differ in respect to, e. g., domain—restricted vs. unrestricted—and the way we evaluate. The corpus used in (Zechner, 2001b) contains excerpts from, e. g., TV-shows and meetings which do not necessarily consist of negotiations. Also, a summary in DiaSumm is based on a subset of the *wording* in the dialogues where the notion of importance is not binary but annotated with a discrete number in the codomain of $[0, 1]$. In our case the summaries ideally include the features the dialogue partners agreed on.

(Zechner, 2001b, section 4.1) contains a black box evaluation of the complete DiaSum system with all its sub-components combined. The first thing we have to do is to understand how Zechners use of *accuracy* as metrics of his evaluation compares to our results. Basically an accuracy for each segment of the summary is computed. This segment-based accuracy is then averaged over the whole dialogue.

$$sa_{s,N} = \frac{summ_s r_s}{\sum_{i=1}^{N} rsort_{s,i}} \tag{4.5}$$

In Zechner's case an important feature of summarization is the size[12] of the final document compared to the original. Compression rate is of minor interest to us since we are interested in the objects being accepted by both partners. In (Zechner, 2001b, p. 87) the size of the summaries, the accuracy and the *gold standard* as shown in table 4.4 are given. Fixed length summaries—"nucleus-IUs" only[13]—are used as human gold standard. Now, the most similar measurement to Zechners summary accuracy we can compute is our recall (definition 4.2). This would mean that we outperform the results of the summary in (Zechner, 2001b) with 11.4% on this relatively small test set. If we take the dialogues pre-processed by our speech recognizers, we receive an average accuracy (recall) of 0.53% for English–English,

[12] The research around summarization in general uses terms, like *compression rate* or *size* to indicate how much of the original input is present in the summary which is an important and interesting information. The size in table 4.4 has merely been included for completeness reasons.

[13] A nucleus-IU is the most important information unit for a topical segment

Table 4.4: Evaluation of the complete system performance of DiaSumm.

sub-corpus	Gold	$sa_{s,N}$	Size (%)
8E-CH	0.709	0.597	13.1
DT-NH	0.791	0.554	20.9
TH-XF	0.764	0.541	11.4
DT-MTG	0.705	0.606	14.9
4E-CH	0.793	0.614	12.9
EVAL-NH	0.850	0.506	14.4
EVAL-XF	0.790	0.566	13.8
EVAL-MTG	0.704	0.583	16.0
Average	0.763	0.571	14.7

0.40% for German–German and 0.65% for the bilingual German–English dialogues. We concede that our summarizer performs worse than DiaSumm for mono-lingual dialogues, but better for bilingual dialogues. However, since Zechner cannot reach 100%—his gold standard reaches 76.3% – the comparison is questionable.

4.5.2 Confabulation vs. Mistake

Next, we make a comment on the evaluation methods used above which is described in, e. g., (Mani et al., 1998). Our main point is that such an evaluation fails to correctly characterize a class of errors which are present in our scenario but not in many others. For a summarizer like the one presented in, e. g., (Marcu, 1998), which selects salient parts of a source document to obtain the summary, it might be reasonable to use the metrics *precision* and *recall* based on **TN, FP, FN** and **TN** as defined above. In such a summary, the result contains *copied* information from the source document only. An evaluation is based on comparing the selected sentences in the summary with those of a gold standard where a binary selection criterion is used: either we selected a member of the gold standard or not. However, there is never the case where something else shows up in the summary other than material from the source.[14]

[14] In what follows we use the following two concepts: **Negotiation Feature**: Negotiation Features are negotiation objects as found in a summary. They are mostly compound objects as shown in, e. g., figure 3.4 (see page 100). **Negotiation Object**: Negotiation Objects are either atomic objects (i. e., strings as values for the role `has_name` or time

In the case of our summaries we may also select wrong, not accepted objects. Additionally we might end up with a summary containing negotiation objects not even uttered by any of the speakers. This is due to the imperfection of our processing steps causing errors beyond *selecting a wrong object.*

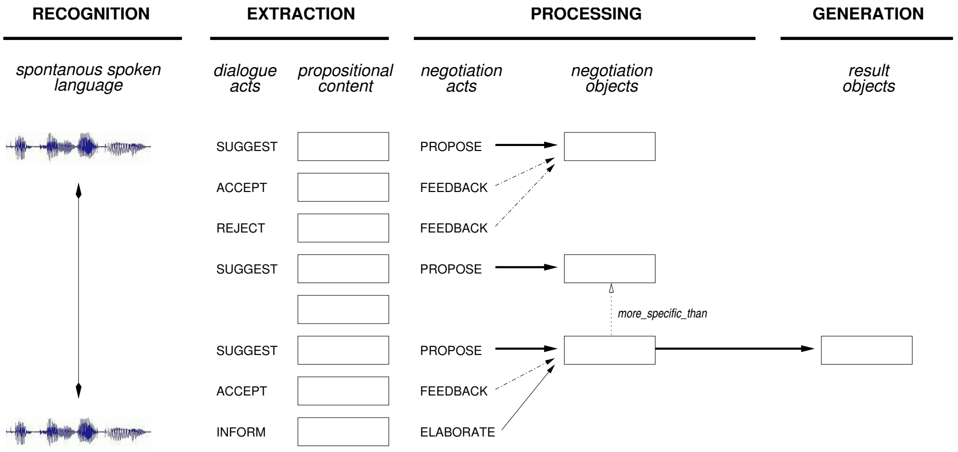


Figure 4.16: Confabulation in summarization. The sources for confabulations are *i*) RECOGNITION: Output from the ASR are almost always incorrect. *ii*) EXTRACTION: The recognition of dialogue act and extraction of propositional content produce errors. *iii*) PROCESSING: The interpretation of the extracted information in context may yield wrong result.

A good example for this phenomenon is shown in the excerpt 55 on page 80 which has been taken from our evaluation corpus where "*When would be a good time for us to meet?*" is recognized as "*one would be a good time for us to meet.*" Here the speech recognizer fails to recognize the spoken utterance correctly and introduces a non-spoken time expression as well as changes the dialogue act. Consequently, the output from extraction will likely be false as well. These kinds of errors might propagate throughout the whole processing chain and we will eventually find negotiation objects in the summary which, in fact, have not been part of the dialogue. We will refer to these objects as *confabulations*[15]. Generally, it is thus imaginable

expressions—tempex) or compounds (i. e., cities). A Dialogue Object might be a Dialogue Feature.

[15] We have searched for a good name for this phenomena. Suggestions were "hallucination", "apparition" and "confabulation." An anonymous native English speaking

that such confabulations will pop up anywhere in the processing chain as indicated in figure 4.16.

We suggest naming the case when we introduce a feature in the summary not present in the dialogue *Confabulation-based False Positive* (in what follows **CFP**). Now, the phenomenon of confabulation is, of course, not distinguished in table 4.3 and indeed not in the definitions on page 164. We start by redefining the standard classifications **TP** ... **FN** and add the definition of **CFP**:

- **True Positive—TP** The Feature approximately corresponds to the gold standard. This means that the feature is either (1) a 100% match, (2) it was not specific enough or (3) too specific[16].

- **False Positive—FP** A feature was not part of the result of the negotiation, but received a non-confabulative feature in the summary.

- **True Negative—TN** A feature was not part of the result of the negotiation, and not included in the summary.

- **False Negative—FN** A feature included in the gold standard *i*) received a wrong non-confabulative value, or *ii*) was not included in the summary.

- **Confabulation-based False Positive—CFP** A feature received a confabulative value whether the feature is part of the gold standard or not.

Next we modify table 4.3 yielding the one depicted in table 4.5 by introducing a confabulated feature, Z, on the summary side.

Note that **CFP** takes events from the false negatives as well as from the false positives. Consequently, the false negatives and false positives get lower, meaning that precision and recall get higher.

Since we have merely cleaned the basis for the formulae 4.1–4.4, the definitions for precision, recall and fallout remain untouched. These metrics now really mirror the intended values. We would, however, like to compute

colleague found the two former alternative suggestions "too amateurish." According to "The Skeptic's Dictionary"—`http://skepdic.com`—the term *confabulation* is used within psychology: "A confabulation is a fantasy that has unconsciously replaced fact in memory. A confabulation may be based partly on fact or be a complete construction of the imagination."

[16] Example of (2) is when the correct date included a time, which was not captured. Example of (3) is when a date with time was annotated but the feature contained just a date.

Table 4.5: Modified version of the Category Task Contingency Table. Additionally to **TP, FP, TN, FN** we introduce **CFP** representing the case where a feature not present in the dialogue has been introduced in the summary due to erroneous processing. X and Y are distinct features actually occuring in the dialogue whereas Z is a distinct but confabulated feature. ϕ represents no feature. **CFP** eventually occurs in two positions, namely in the case where feature Z is a confabulation-based error.

	Gold Standard	
Summary	X	ϕ
X	**TP**	**FP**
Y	**FN**	**FP**
Z	**CFP**	**CFP**
ϕ	**FN**	**TN**

a value indicating how confabulative our system is, and we do this by a formula related to the one of fallout:

$$\mathit{confabulation} = \mathit{CFP}/(\mathit{CFP} + \mathit{TN}) \tag{4.6}$$

Additionally, we have to redefine the definition of fallout:

$$\mathit{Fallout} = (\mathit{FP} + \mathit{CFP})/((\mathit{FP} + \mathit{CFP}) + \mathit{TN}) \tag{4.7}$$

Furthermore, we introduce two formulae for computing the *relative confabulation* (**RC**) and the *total confabulation* (**TC**) in which the number of confabulations are related with the other errors (Equation 4.8) and the number of confabulations with the total number of features in the summary (Equation 4.9):

$$RC = \frac{\mathit{CFP}}{\mathit{CFP} + \mathit{FP} + \mathit{FN}} \tag{4.8}$$

$$TC = \frac{\mathit{CFP}}{\mathit{CFP} + \mathit{FP} + \mathit{FN} + \mathit{TP}} \tag{4.9}$$

The evaluation of our system using the new metrics will be a very cumbersome task and will require methods not unlike those proposed for *glass box* evaluation (Simpson & Fraser, 1993). Nevertheless, we indicate how such an evaluation could be performed: For an evaluation of our summarizer including confabulation, the evaluator has to do the following steps while setting the gold standard:

1. Trace every dialogue object introduced in the dialogue.

   For each object do:

   (a) keep track of where in the dialogue the object was introduced.

   (b) keep track of which effects on the context the new object has, i. e., which (eventually compound) dialogue objects could have been created by introducing the new dialogue object.

2. Select the agreed-upon objects (negotiation features) for the summary.

The dialogue is now run through the system and the summary computed. Then, the usual evaluation is performed with one exception. To be able to distinguish the confabulation-based errors (**CFN**) from other dialogue objects from the dialogue—**FN** and **FP**—the evaluator has to additionally compare each feature not in **TP** in the summary including its source positions in the dialogue with the list of dialogue objects actually occurring in the dialogue.

Clearly, such an evaluation is not realistic for more than just a few dialogues, so we have re-evaluated the five dialogues used for the comparison in the previous section. The basis for the evaluation was the dialogues processed by one of our speech recognizers. The result is depicted in figure 4.17.

As can be seen, our thesis turned out to be true! **RC** = 17%, i. e., on average, 17% of the false positives and false negatives belong to the CFP class. The total confabulation (**TC**) is 12%, i. e., a bit more than every tenth object in the summary was not licensed by the original dialogue. In the next section we dissect the dialogues in an effort to find the loci of the confabulations.

4.5.3 Error Analysis

In the beginning of section 4.5.2 we predicted that errors in some of the processing steps would make it all the way to the summary. To test this hypothesis, we carefully dissected the 5 evaluated dialogues to see if our thesis could be proven. As can be seen in figure 4.17, we found several errors thus proving that confabulation errors actually exist. In our system the errors have different loci as shown below. We additionally give some of the most typical examples.

- **Speech recognition errors** In this example, one of the speakers suggested booking a hotel called "*Maritim*." Instead, the bigram "*Variete*

Speech recognized						
Dialogue	1	2	3	4	5	avg
Turns	43	33	50	52	52	46.00
TP	5	8	9	6	10	7.60
FP	3	0	7	8	6	4.80
CFP	3	0	5	2	6	3.20
TN	85	89	71	79	75	79.80
FN	8	7	12	9	7	8.60
Precision	0.62	1.00	0.56	0.43	0.62	0.65
Recall	0.38	0.53	0.43	0.40	0.59	0.47
Fallout	0.03	0.00	0.09	0.09	0.07	0.06
f-score	0.48	0.70	0.49	0.41	0.61	0.54
Rel. Conf.	0.21	0.00	0.21	0.11	0.32	0.17
Tot. Conf.	0.16	0.00	0.15	0.08	0.21	0.12

Figure 4.17: Evaluation of five German–German dialogues, manually transcribed and processed by speech recognition.

im" was predicted by the speech recognizer. The German word "Variete" means "vaudeville", so our shallow analysis component took this for a suggestion concerning entertainment resulting in the following negotiation object selected for summary generation:

```
ENTERTAINMENT (P51*)
  HAS_LOCATION --> NONGEO_LOCATION (P52*)
                     HAS_NAME=variete
```

Another example of this kind of error is "Michelle" in the summary of dialogue 99_33 in the appendix. There, the spoken chain "shall we travel on the first of March. . . " is recognized as "Michelle we travel on the the first of March. . . "

- **Shallow Analysis** In this example we provide still more proof for the well known fact that the precision of a shallow analysis component has its limitations. Segments that are especially long are hard to analyze correctly. But shorter segments give rise to errors found in the summary too.

  The German expression *Geschäft ist Geschäft* which translates to *business is business* is misinterpreted as a suggestion of a location. This is because *Geschäft* has another reading: *Store* or *Shop*. The consequence is that the following appointment is found in the summary:

```
APPOINTMENT (P16*)
  HAS_LOCATION --> NONGEO_LOCATION (P10****)
                     HAS_NAME=geschaeft
```

- **Inferences** The algorithms for adding context information to new utterances—the *complete* operation and the *path of instant completion*—make mistakes. This is mostly because the wrong anchor for the operation is found. Therefore, possibly partly confabulated objects are constructed with either
  - **wrong time expressions** A common error was that the anchor of the new time expression was at the end of a month, e. g., June, and the new time expression consisted only of a day-of-week at the beginning of the month but without mentioning the month explicitly, e. g., "the third." The inference algorithm then chose the month of the nearest time expression yielding "the third of June."
  - **wrong negotiation objects** When completing the new propositional content with the context, the wrong anchor was chosen, sometimes giving birth to objects which were partly correct, partly incorrect.

4.5.4 Discussion

Evaluations metrics related to the ones presented above has been suggested in slightly different settings and areas. A common measurement for speech recognizers is the *word accuracy* (WA). WA is computed based on *deletions*, *insertions* and *substitutions* of words in the original spoken utterance:

$$WA = 100\left(1 - \frac{W_S + W_I + W_D}{W}\right) \tag{4.10}$$

For the evaluation of dialogue systems some researchers have gone one step further and defined *information content* (IC) (Simpson & Fraser, 1993) and *concept accuracy* (CA) (Boros et al., 1996; Baggia et al., 1999). These measures are based on the counting of the recognized semantics given an utterance; instead of counting insertions, deletions and substitutions on the word level, these are counted at the semantic level. As a basis for the computations, Boros et al. use a list of *semantic units* represented as a list of attribute–value pairs. An interesting observation in (Boros et al., 1996) is that CA goes hand-in-hand with the WA.

The computation of CA is, however, not straight forward. For some systems, like the sundial system, it might be sound to use the strategies above, but for more structured representations, the difference between structures are not straight forward to compute. A possibility could be to use a scoring mechanism as the one proposed for OVERLAY (see section 3.7.4).

Another aggravation is that WA and CA are, however as they stand too less informative. Again, consider the excerpt 55 on page 80:

> When would be a good time for us to meet?
> One would be a good time for us to meet.

If we compute WA for the sentence, there is one substitution: $WA = 90\%$ which can be viewed as a good result. However, the single substitution (When → One) has as a consequence that not only is the propositional content wrong, but also the dialogue act. Thus the (negative) effect of a single substitution during speech recognition is comparatively much worse than if, say "for" would be deleted. Still the WA is the same for these errors. Therfore we beleive that more fine grained metrics are needed which take phenomena like these into account.

In (Simpson & Fraser, 1993), the concept and usage of *black box* and *glass box* evaluation for dialogue systems is discussed. By a *black box evaluation*, the authors refer to an evaluation where the behavior of a system is evaluated without knowing anything about the inside of the system. The evaluator is then observing the input–output of the system. For the *glass box evaluation* however, access is given to the different modules of the system, and each processing steps is evaluated.

The fundamental difference between our evaluation and the one based on CA only, is the general approach taken here. Instead of focusing on the recognition of semantics for non-hierarchical attribute–value pairs during parsing, we take any error in the processing chain into account. The drawback is, however, that the efforts necessary for our evaluation are much higher. Still, in our case we have not performed a true glass box evaluation but merely something in between. Since, for instance, the computation of CA requires a semantically annotated corpus, the resources needed for a glass box evaluation are, indeed, considerable.

4.6 Conclusion

In this chapter we presented an application—*multilingual summary generation*—based on the functionality of the dialogue component in VerbMobil. By reusing already existing components, e. g., transfer and generation, where the latter was extended with a formatting functionality, we described how the missing link—the summary generator—functions. The complete generation chain consisting of the following steps was presented and discussed:

- **Content Selection** The most *recent special accepted negotiation objects* were selected from the dialogue memory. This process returned a quadruple of negotiation objects according to the four negotiation topics *negotiation*, *traveling*, *accommodation* and *entertainment*. Together with information about speakers, location and date, the negotiation objects prescribe the content of the summary.
- **Summary Generation** A bottom-up generation algorithm inspired by (Marcu, 1997), working in a similar way as constraint satisfaction algorithms used in, e. g., (Becker et al., 1998). The generator maps the negotiation objects to sequences of VITs which are made coherent within each topic.
- **Multilinguality** By making use of the transfer component we could, in principal, generate summaries in *any* language completely incorporated into the system.

We compared the performance of the summary generator with DIASUMM, one of the few systems summarizing spoken negotiation dialogue. Finally, we carefully evaluated our system using a number of testsets consisting of mono- as well as bi-lingual dialogues, manually-transcribed recognized by the speech recognizer. We showed that the standard evaluation method for summarization is inadequate in that it is not capable of capturing errors due to erroneous recognition and/or processing. Hence, we introduced a new concept we call *confabulation* which characterizes our summarizer by providing a numeric value representing how hallucinative the summary machine is. In our system, errors are not only caused by selecting the wrong features or items for the summary but also because of errors in, e. g., speech recognition and during inference. A dissection of the errors in the last evaluation was performed and we showed some typical confabulative errors.

Chapter 5

Conclusion

New ideas and the combination of different approaches—manually constructed knowledge bases as well as the result of applying learning methods to manually annotated corpora—laid the basis for the success of the dialogue module. Instead of focusing on algorithms providing high precision for a few examples, robust methods for the processing of massive amount of data have been developed. This has been possible because of the access to great amount of data which has been transcribed and annotated with different linguistic information.

Our first aggravation while designing our module has been the nature of human–human spontaneously spoken speech. Performance phenomena, like hesitations and deliberations make the output from speech recognition and dialogue act recognition far from perfect. In fact, with speech recognition with a word recognition accuracy of 80% every *fifth* word is wrong. Section 3.5.1 reports on a recognition rate for dialogue acts of between 60 and 80%. Still, some of the tasks of a discourse module within a mediating scenario are the same as for a human–machine scenario. For instance, context-dependent interpretation in VerbMobil is performed much in the same way as in SmartKom. The main difference is that, contrary to the SmartKom case, in VerbMobil, there exists no action planner which controls the dialogue and hence can *dynamically* predict in more detail what is going to happen next. Instead, the interpretation process, especially for short utterances, is guided by a *static* set of templates in VerbMobil.

Another peculiarity of our dialogues is the difference in turn complexity between human–human dialogue and man–machine dialogue. Although, almost 70% of the turns in our corpus consist of one (42.6%) or two (26.8%) segments—the average number of segments per turn is 2.1—there are many turns with four, five or even six segments. Looking at content, we find that, while the core intention of a turn may be the same, e. g., rejecting a suggestion, some interlocutors produce additional linguistic material, e. g., due to rules of politeness. Such phenomena are not found in human–machine dialogue. Furthermore, the way humans speak to each other in one language differs from that of communicating via a human interpreter or VerbMobil. For interpreted dialogue, we almost exclusively find turns consisting of one or two segments, i. e., the intentions are verbalized in a short and concise manner. Monolingual dialogue includes more spontaneously spoken speech where phenomena, such as deliberations and thinking-aloud are common findings. For the task of maintaining the intentional structure, we encounter rather big differences between the mediating role of a dialogue module in VerbMobil and the controlling role of a dialogue module in a human–machine dialogue system. The main reason is, again, that the structure of man–machine dialogue tends to be very simple whereas the human–human dialogue is more complex. VerbMobil's negotiation scenario is one of the main reasons for this: in VerbMobil, every turn—except possibly the first—has at least a backward looking function. Likewise, every turn—except possibly the last—has a forward looking part.

In the running system, our processing is based on a segmentation of the turns into segments represented by intention (dialogue act) and propositional content (instances of our domain model). By choosing this approach, i. e., skipping much of the information provided by, e. g., an HPSG grammar, we sacrifice detailed linguistic information in favour for coarse-grained one; the goal being robustness against, e. g., performance phenomena common in spontaneously spoken speech and errors during speech recognition. Although the information is less precise with our approach, it is possible to assign a meaningful interpretation to heavily distorted utterances. Our less sophisticated information structures allow for flexible processing. This flexibility is necessary because of the lack of control of the dialogue.

A dialogue manager operating within a translation system has as main task to support translation. However, we have shown that, with minor extentions to the discourse processing, the frame-based approach to the representation of propositional content provides a good basis for additional functionalities, such as, generating summaries mirroring the result of the negotiation.

In the following sections we conclude the main contributions of our thesis:

The Intentional Structure

The theoretical part of this work has been inspired by, for instance, (Bunt, 2000; Chu-Carroll & Brown, 1998; Carletta, 1996; Allwood, 2000; Levinson, 1983). The basic assumption has been that cooperative negotiation dialogues can be represented with a tree-structure. Our intentional structure consists of several layers representing different abstractions of dialogue. Our layered approach supports modularity and enhances robustness. There, our usage of *moves* is novel. Still, it is a natural consequence of the finding that different speakers, possibly from different cultures, verbalize their core intention very differently.

During runtime, the structure is maintained using a combination of handcrafted and semi-automatically acquired knowledge. Our plan processor deploys a top-down expansion strategy with chronological backtracking which, in combination with memoization techniques, has turned out to be efficient and fast. Object-oriented programming techniques further enhance the robustness.

Complete → Overlay = Default Unification + Scoring

The formalization of the completeness algorithm (see section 3.7.3 and 3.7.4) provides a precise description of how a maximum amount of consistent context information can be added to new information. The assimilation operation allows old information from partly incompatible ones to be inherited. In addition to the purely structural computations performed by OVERLAY, a formula for the computation of a score mirroring the resemblance of the two structures was given. The usefulness of the OVERLAY operation has successfully been demonstrated not only in succeeding projects, e. g., SmartKom (Reithinger et al., 2003; Wahlster, 2003), where it functions as the basic operation for inheriting contextual information (Pfleger, Alexandersson, & Becker, 2003; Pfleger, Engel, & Alexandersson, 2003; Alexandersson & Becker, 2003b, 2003a) but also in the MATCH project (Johnston et al., 2002).

The Summary Generation Algorithm

In a broader perspective, our approach to multilingual summary generation is based on the presence of the transfer and the generation component (VM-GECO) in the VerbMobil system. Contrary to, e. g., (Stede, 1999) we base our summary generation on German summaries and let the transfer component assist in translating the summaries on a semantic level in case a summary other than German is requested. This has the advantage that we can keep the complexity of our generator and its knowledge sources relatively small.

We base our generation of multilingual summaries on the content of the discourse memory. There, we encountered that summary generation in our case differs from many other generation tasks in one important aspect: after having selected appropriate parts of the discourse memory, the generator has to verbalize the whole data base. This approach has been used by other researchers before, i. e., (Marcu, 1997). Our solution is a data-driven bottom-up generation algorithm which resembles constraint satisfaction techniques much in the same spirit of (Becker et al., 1998; Becker & Löckelt, 2000).

Evaluation and Confabulation

Most approaches to evaluation are based on the assumption that precision, recall can be computed based on the *true/false positive/negative.* These four different cases are found while evaluating our dialogues as well. There is, however, another peculiarity in our case: during some of the processing steps, "items" not part of the actual dialogue may be introduced into the processing. These items may even appear in the summaries. In a sense, these items have not even been mentioned but "hallucinated" by the system and thus we have called such items *confabulations.*

We sophisticated the standard classifications used for evaluating, for instance the performance of summarizers (*true positive* etc.) by introducing the case *confabulation-based false positive* for the cases where, in our case, something appeared in the summary which was not mentioned in the negotiation but a product of erroneous processing. Related to that, we introduced two measures stating how "confabulative" a system is, i. e., mirroring how big part of the errors stem from confabulations.

5.1 Main Scientific Answers

Before we give a list of future research we provide answers to the the scientific questions posed in the introduction (page 15):

Representation issues We model utterances with intention (dialogue act) and propositional content (instances of our domain model). With this assumption, we have focused on two points:

Tracking Propositional Content A fundamental task for every dialogue manager is to provide a context-dependent interpretation for communicative acts performed by the user. In our case, this is achieved through a combination of templates which map the input structure onto a negotiation object. The negotiation object is then merged with the context using an default-unification-like operation to obtain the context-dependent interpretation.

Dialogue Games In negotiation dialogues, a contribution from one speaker contains a backward-looking act and optional a forward-looking act. These acts are reactions and/or attitudes towards discourse objects under negotiation, e. g., initiatives (suggestions) and feedbacks (rejections) etc. These basic attitudes, called *moves*, form *negotiation games*. However, depending on, for instance, setting and culture, a speaker conveys the basic moves in different ways. On a more abstract level, the dialogue is described by *dialogue phases* such as introduction, negotiation and closing.

Controlling versus mediating Mediating a dialogue between two interlocutors means that the dialogue module has no control over the course of the dialogue. Whereas a human–machine dialogue system can perform clarification dialogues, e. g., to resolve ambiguities or correct erroneously recognized and interpreted user contributions, an eavesdropping system must be flexible and has to be able to follow the dialogue no matter how strange it might look like. In particular, interpretation has to be robust, and we advocate the use of supervised acquired n-gram language models for the recognition of dialogue acts and dialogue moves instead of methods based on hand-crafted knowledge. Semi-automatically acquired knowledge has an important role in the success of the system and has only been possible because of a large corpus annotated with respective information. We use a relatively coarse grained representation—instances of an ontology tailored towards the domain instead of detailed more traditional linguistic semantic representation formalisms—in combination with robust processing algorithms. Such an approach is forgiving towards errors in speech recognition.

Spontaneous speech Some of the recognition errors due to spontaneous speech are very hard to circumvent. Recognition errors causes the translation process to fail. However, on the one hand utilizing the fact that we know what to expect and, on the other hand, relying on flexible robust methods, such as hidden markov models (HMMs), we are still in the position of predicting what course the dialogue has taken.

Multilinguality Multilinguality partly means multiculturality. Multiculturality has a similar effect on the characteristics of the dialogues as the setting; multilingual dialogue interpreted through a machine is simpler in structure than monolingual dialogue. Multilingual dialogues differ in structure particularly how interlocutors from different cultures convey their intentions. For the modeling and processing, the abstraction level *dialogue moves* is essential for dealing with this phenomenon.

Minutes and Summarization We have shown that discourse management based on AI techniques like description logics for the modeling of our domain in combination with default unification like operations and dialogue acts for modeling attitudes put us in the position of computing the result of the negotiation. At the end of the negotiation, the dialogue memory contains, amongst other things, those objects agreed-upon. After the selection of the interesting objects—the most recent special objects—of the dialogue memory, we have shown that the generation process gains in flexibility and completeness if it is based on the insight that *everything* has to be generated. This has moved our interest into constraint-satisfaction techniques and bottom-up generation. Utilizing the presence of a generation and a semantic transfer module in the VerbMobil system, we are able to produce summaries in all VerbMobil languages.

Evaluation Summarizing based on comparatively knowledge intense methods unveils that the standard evaluation metrics as described in, e. g., (Mani et al., 1998) are not capable of explaining all errors. In particular, since some of the processing steps introduce discourse objects into the summaries not mentioned at all in the negotiation, it has been necessary to introduce *confabulation* for a correct evaluation of the summarization functionality.

5.2 Future Work

Many of the results from the VerbMobil project have been further developed. In particular, we have provided the formalization of the completion algorithm and named it "overlay." Overlay has been used as the basic operation for discourse processing in the SmartKom project (Reithinger et al., 2003). In SmartKom, we have enhanced discourse model with, for example, a double-threaded focus tracking and a three-tiered discourse model (Luperfoy, 1992).

Fusing Discourse Modeling and Plan Recognition

For the task of actively participate in dialogue as well as for tracking dialogue, the plan processor—or the action planner deployed in SmartKom—should be more tightly coupled with the the dialogue processor (discourse modeller in SmartKom). This will be explored in the upcoming AMI project concerned with meeting room recordings much in the spirit of (Morgan et al., 2002; Waibel et al., 2001).

Overlay and sets

Ongoing work includes extending overlay to work also with set-like structures. The outcome of overlaying a set of some entities with another set depends on many different aspects. Some of the more prominent are type of entities and function in context.

Automatic generation of minutes

Our efforts for the generation of minutes provides a good basis for further research. The basic idea will be further developed in the upcoming AMI project where, similar to the meeting room project, a (multimodal) meeting browser will be developed supporting, e. g., a seamless smooth transition going from word recapitulation via minutes to summaries.

Annotation of moves and games

Our framework around the intentional structure—including moves and games—has to be verified on other types of dialogues. The upcoming AMI provides a good opportunity for doing that.

The Summary Generation Algorithm

Our implementation of the generation algorithm is an imperative solution where a hand crafted search algorithm is pursued. Future work will include a reimplementation following the ideas suggested in (Becker & Löckelt, 2000; Scheffler, 2003) or (Kay, 1996; Koller & Striegnitz, 2002; Carroll, Copestake, Flickinger, & Poznanski, 1999).

Appendix

A sample dialogue act definition

SUGGEST

Upper level dialogue act: PROMOTE_TASK
Dialogue phase: NEGOTIATION
Related propositional content:
contains the suggested proposition, e.g. a date or duration, a location, a selection of transportation or accommodation, an action

Definition:
With an utterance expressing a SUGGEST the speaker proposes an explicit instance or aspect of the negotiated topic (not neccessarily only one instance or aspect, could also be a set of instances). A further point of the definition is that the proposed instance must be either a new referent or a further specification of an already introduced one.

German Example: *cdromx ,moko*

```
m067arr1_035_LAU_000000:  <*tGER> ja , das w"are doch gut ,
(ACCEPT) <#Klopfen> dann k"ummern Sie sich um den Flug und
dann kann ich mich um die Hotelbuchung noch% k"ummern .
(SUGGEST)                                                □
```

German Example: *cdrom14 ,j521a*

```
CLS003:  <Schmatzen> <A> ja , ich schau' hier auch grade .
<Schmatzen> (DELIBERATE) <A> <"ahm> es w"urde +/Ende/+
Ende Januar <P> vom achtundzwanzigsten<Z> bis zum zweiten
Februar bei mir gehen <#Klicken> .  (SUGGEST)            □
```

German Example: *cdrom14 ,j511a*

ULP009: <;T>ja , ich "uberlege <:<#> soeben:> . (DELIBERATE) <A> vielleicht <"ah> ab dem zehnten<Z> Februar w"are es <:<#> m"oglich:> bei mir <#Klicken> . (SUGGEST)

JMP010: <A> <"ahm> tut mir <:<#Klopfen> leid:> ,(REJECT) liegen zwei<Z> <A> <:<#> Gesch"aftsreisen:> bei mir vor , (GIVE_REASON) <:<#> einmal nach Bremen und Minden:> . (INFORM) <A> ich k"onnte <:<#> ab:> sechzehntem Februar . (SUGGEST)

ULP011: <A> das ist bei +/wi=/+ <h"as> mir nun wieder schlecht. (REJECT) <"ahm> da geht es fr"uhestens am achtzehnten Februar , <#> dem <:<#> Sonntag:> <#> . (SUGGEST) □

here a set of possible options is suggested

German Example: *cdrom7 ,m068n*

JUJ018: ja , das kommt mir auch sehr gelegen . (ACCEPT) lieber an dem Samstag oder lieber an dem Sonntag ? (SUGGEST)

MCE019: <"ah> mir w"ar' der Samstag lieber . (ACCEPT) □

English Example: *cdrom13 ,r005k*

CK2001: <A> okay <;period> <;seos> what date would be good <;period> <A> <;seos> (REQUEST_SUGGEST)

PN1002: <#Klicken> <A> almost any day <;comma> as long <;comma> as it <#> is not the weekend <;period> or Wednesday <;period> <#Klicken> <;seos> (SUGGEST)'' □

A sample move definition

initiative

Forward/Backward An initiative has always a forward looking aspect since it introduces a new topic or aspect. There is, however, one exception: The initiative has a backward looking aspect when it follows a transfer-initiatiative.

Definition This move covers the cases where a new proposal or topic is introduced which opens up a new discourse segment. Cases:

1. When a new proposal or topic is introduced (see example 2.)
2. When a suggestion refines a previous proposal or topic that has been accepted explicitly (Example 3)
3. When a direct counter proposal is made (see example 4).
4. When the speaker commits him-/herself to do one or more specified action(s) (see examples 8 and 9).

Note When a turn has both a descriptional and a propopositional character, but can not be segmented into a describe and an initiative part it should be labeled with an initiative.

Example 2 (cdrom15par: g015ac)

ABE001: greet	GREET -	Gott , Herr Sauer . <A> *(Hello, Mr Sauer)*
ABO002: initiative	INIT -	ja<Z> , wir wollten eine<Z> Ge<Z>sch"aftsreise vereinbaren "uber eineinhalb Tage. <A> *(Well, we wanted to schedule a business trip for one and a half days)*
ABE003: clarify-query	REQUEST_CLARIFY -	ach , das war die nach Hannover , richtig<Z> . *(oh, that's the one to Hanover, right?)*
ABO004: clarify-answer	FEEDBACK_POSITIVE -	genau . *(exactly)*
	CLARIFY -	zwischen Juni und August . <A> *(between June and August)*

□

Example 3 (cdrom15par: g003ac)

AAJ021: response

- REJECT - n<Z>ein , geht nicht , *(no , impossible)*
- GIVE_REASON - da hab' ich um vier-zehn Uhr einen Termin <A> in Starnberg . <A> *(I have an appointment at 2 pm at Starnberg)*

AAJ021: initiative

- SUGGEST - <hm> <A> <Schmatzen> ja , aber am Mittwoch , Donnerstag , Freitag , sechzehnter , siebzehnter , acht-zehnter , das g<Z>eht . *(Well, but on Wednesday, Thursday, Friday, 16th, 17th, 18th is possible)*

AAK022: response

- INFORM - da bin ich ganz frei<Z> , *(I'm free then ,)*
- ACCEPT - das pa"st mir hervorragend11@ . *(that suits me perfect .)*
- **AAJ023:** initiative
  - SUGGEST - <@11<hs> machen wir's gleich am Mittwoch , am sechzehnten12@ ? *(let us do it directly on Wednesday , on the 16th ?)*
- **AAK024:** response-initiative
  - ACCEPT - @12ja *(yes)*
- **AAK024:** initiative
  - SUGGEST - , <@12<A> Mittwoch und Donnerstag , wenn<Z> n"otig k"onnten wir sogar auf Freitag verl"angern . <A> *(Wednesday and Thursday , if necessary we can extend it to Friday)*

□

Example 4 (cdrom15par: g010ac)

ABD023: initiative

- COMMIT - dann k"ummer' ich mich<Z> <P> um die Tickets . <P> *(I take care of the tickets)*
- BACKCHANNEL - gut7@ . *(good)*

ABA024: initiative

- INIT SUGGEST – @7ja @7vielleicht k"onnen wir auch in Hannover noch irgendwas<Z> unternehmen ? *(maybe we could do something in Hanover to ?)*

□

Example 5 (cdrom15par: g009ac)

ABA042: initiative	SUGGEST -	ah , ich glaub' , da ist Hannover nicht so<Z> "uberragend , *(well, I believe that Hanover is not that good)*
	INFORM -	da nehmen wir lieber ein Museum . *(Let us go to a museum .)*
ABD043: response	ACCEPT -	gu<Z>t .
	INFORM -	<A> ja , ich nehm' an , wir kriegen da dann dort die Karten . <P> *(well, I guess we will get the tickets there)*
	INFORM -	braucht12@ man12@ sich12@ nicht12@ drum12@ mmern12@ . *(we don't need to worry)*

□

Example 6 (cdrom15par: g009ac)

ABA018: initiative	INIT SUGGEST –	ja , am n"achsten Tag<Z> <hm> dann zur"uck . <A> *(Well, back the next day)*
	SUGGEST -	<hm> <A> <hm> ja , da k"onnen wir eigentlich dann ja schon mittags fliegen , *(we could fly at noon)*
	SUGGEST -	oder wir k"onnten nat"urlich auch noch den<Z> Nachmittag nutzen , um in Hannover irgendwas zu unternehmen ? *(Or we could us the afternoon to undertake something)*
ABD019: response	ACCEPT -	ja , *(yes)*
	ACCEPT -	gerne , *(sure)*
	ACCEPT -	machen wir das *(Let us do that)*
	INFORM -	doch . da gibt 's doch Einiges zum anschauen . *(yes, there are a lot of thing to see there)*
ABA020: initiative	SUGGEST -	und dann <A> <P> k"onnten wir so<Z> gegen acht oder gegen neun zur"uckfliegen . *(and then we could fly back around 8 or 9 pm)*

□

Example 7 (cdrom15par: g014ac)

ABO016:
initiative

FEEDBACK_POSITIVE - die sind ganz gut , *(they are quite good)*

SUGGEST - Parkhotel , Hotel Cristal , da gibt ’s Sauna und Solarium . <A> *(Parkhotel , Hotel Cristal , they have sauna and solarium)*

□

Example 8 (cdrom15par: g002ac)

AAK047:
initiative

COMMIT - +/@20ich mach’/+ ich mach’ Reservierungen *(I’ll make the reservation)*

COMMIT - f"ur diese beiden Z"uge , +/kauf’ die/+ besorg’ die Fahrkarten , <A> *(for these train trips I procure the tickets)*

DEFER - wenn<Z> das l"anger wird k"onnen wir das ja sp"ater +/noch<Z>/+ noch "andern , vor Ort . *(if it take a while we could change it there)*

AAJ048:
response

ACCEPT - ja<Z>21@ . *(yes)*

□

Example 9 (cdrom15par: g008ac)

AAK061:
initiative

ACCEPT - das kann ich machen , *(I can do that)*

ACCEPT - ja<Z> . <A>30@> *(yes)*

COMMIT - da<Z>30@ beauftrag’ ich meine Sekret"arin . *(I’ll tell my secretary)*

AAJ062:
response

ACCEPT - @30oh @30ja , *(oh yes)*

ACCEPT - @30prima . *(great)*

□

A sample game definition

Negotiation

This dialogue game mostly encompasses the move *initiative* from speaker A followed by a *response* move by speaker B. Optionally speaker B opens the game with an initial *transfer_intiative*. The game response can also be confirmed by speaker A. A special case is the response_initiative: Some contributions from one speaker may function as a response *and* as an initiative at the same time.

However, it can include different negotitating games as well that relate to the appointment scheduling task. It commonly encompasses the moves: initiative, transfer_intiative, response, describe, response_initiative and confirm.

Example 10 (Cdrom39multi-par: moo2e.gam)

negotiation	**AB:** initiative	SUGGEST -	`okay I am free between the twenty second and the twenty fourth`
		REQUEST_COMMENT -	`what is your schedule like?`
	BA: response	ACCEPT -	`ja am vierundzwanzigsten habe ich einen freien Termin` *`(I have an appointment free on the 24th)`*

□

Example 11 (Cdrom23-par: e004ac.gam)

negotiation	**BA:** transfer-initiative	EXPLAINED_REJECT -	`I am going off on vacation then for three weeks so until the first week of December`
	AB: initiative	FEEDBACK_POSITIVE -	`all right`
		SUGGEST -	`I am away the beginning of that week but I could do it towards the end of the week or the beginning of the next week`
	BA: response	ACCEPT -	`okay`
		ACCEPT -	`the beginning of the next week sounds good`

□

Example 12 (Cdrom15-par: g002ac.gam)

negotiation	**BA:** initiative	SUGGEST -	oder die Woche danach? *(or the week after that?)*
	AB: response	SUGGEST -	Montag Dienstag Mittwoch ein-und *(Monday, Tuesday, Wednesday <?>first)*
	BA: response	SUGGEST -	Dienstag Mittwoch *(Tuesday, Wednesday)*
	AB: clarify-query	REQUEST_CLARIFY -	Dienstag Mittwoch? *(Tuesday, Wednesday?)*
	BA: clarify-answer	FEEDBACK_POSITIVE -	ja *(yes)*
	AB: response	ACCEPT -	das w"urde gehen *(that is possible)*
		ACCEPT -	hervorragend *(splended)*
		ACCEPT -	ja *(yes)*

□

Corpus Characteristics

To provide a picture of the distribution of dialogue acts per turn in our corpus, we have counted the dialogue acts for each turn for our annotated dialogues. Basis for the curves are 794 German–German, 375 English–English and 403 Japanese–Japanese dialogues. Three curves are provided: The first (i. e., figure 1) includes all turns, the second all turns including the dialogue acts GREET, INTRODUCE and BYE, i. e., the upper graph in figure 2. The third describes the length of all negotiation turns, i. e., turns not containing the dialogue acts in the second counting. An example of this is the lower graph in figure 2.

Additionally we provide the same information for a small set (10 dialogues) of German–English multilingual dialogues (see figure 3). The figures there are—due to the low number of annotated dialogues—to be viewed merely as a hint than as a finding.

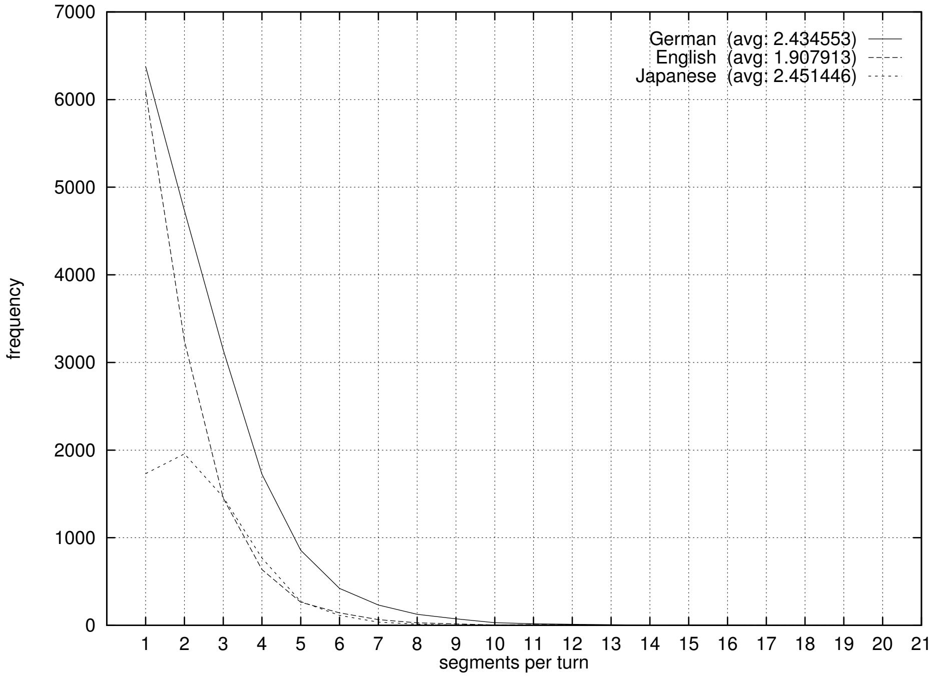


Figure 1: Numner of dialogue acts per turn for all turns for three sets of monolingual dialogues (German–German, English–English and Japanese–Japanese).

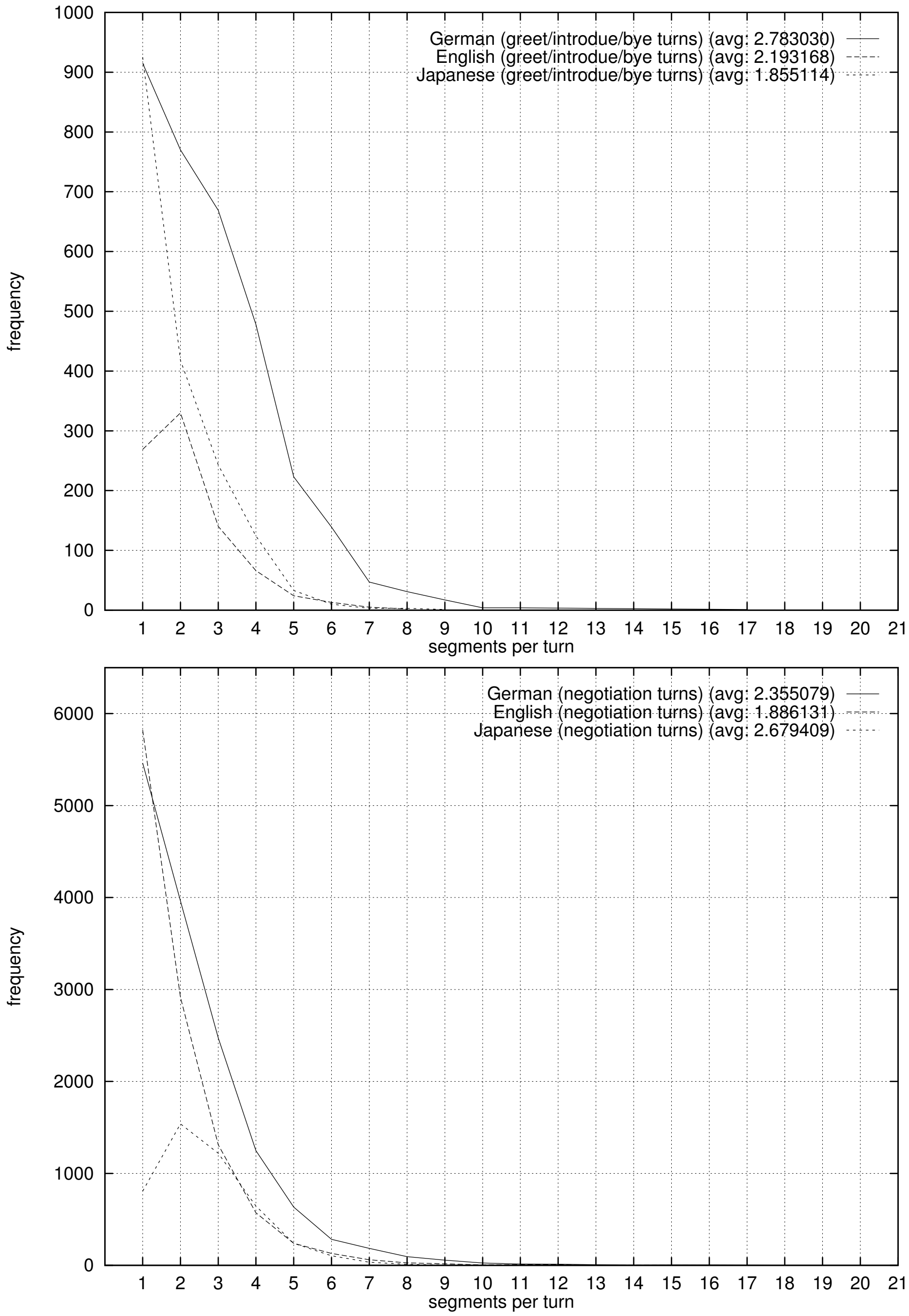


Figure 2: Number of dialogue acts per turn for three sets of monolingual dialogues. The upper figure describe turns containing the dialogue acts GREET, INTRODUCE and BYE whereas the lower turns not containing these dialogue acts.

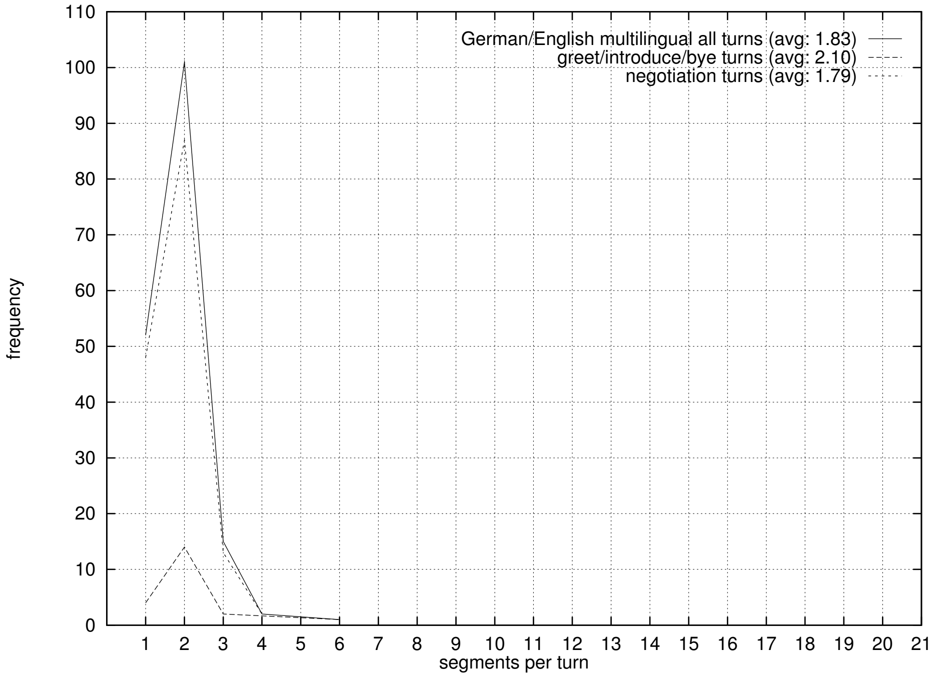


Figure 3: Number of dialogue acts per turn for turns containing the 10 multilingual German–English dialogues. Three curves are given: one for the turns containing the dialogue acts GREET, INTRODUCE and BYE, one for all other turns and, finally, one all turns.

Two Sample Dialogues

We provide our sample dialogue—*Reinhard4*—annotated with propositional content and dialogue act as processed in the VERBMOBIL system. The trace is followed by the content of the dialogue memory and the corresponding German summary.

```
;*************************************************************
; Dialog: 3000/reinhard4
; Time  : NIL
;*************************************************************

; -  sch"onen Guten Tag
;    ^ t1000ageb2e3r1geySHALLOW
;       (((GREET . 30.064) 10.021) ((POLITENESS_FORMULA . 40.993) 13.664))
;       [GREET,any_topic]

; -  hello this is <UNK:Surname..Thompson> speaking
;    ^ t1001benb3e4r1genSHALLOW
;       (((GREET . 54.993) 10.998) ((INTRODUCE . 61.868) 12.373))
;       [GREET,any_topic,has_addressee:[person,
         has_last_name='<UNK:Surname..Thompson>']]
; -  hello hello Mr <UNK:Surname..Schneider>
;      t1001benb4e5r1geySHALLOW
;       (((GREET . 51.717) 12.929) ((INTRODUCE . 60.128) 15.032))
;       [GREET,any_topic,has_addressee:[person,
         has_last_name='<UNK:Surname..Schneider>']]

; -  ja es geht um das Gesch"aftstreffen in Hannover
;    ^ t1002ageb5e6r1genSHALLOW
;       (((INIT . 58.054) 7.256) ((INFORM . 65.007) 8.125))
;       [INIT,scheduling,has_appointment:[appointment,has_meeting:[meeting,
;        has_name='geschaeftstreffen'],has_location:[city,
         has_name='hannover',has_loc_spec=in,has_det=unknown]]]
; -  das ist ja am zwanzigsten Januar um elf Uhr vormittags
;      t1002ageb6e7r1geySHALLOW
;       (((SUGGEST . 71.321) 7.132) ((ACCEPT . 77.315) 7.731))
;       [SUGGEST,uncertain_scheduling,has_date:[date,
         tempex='tempex(ge_2920_0,[from:[dom:20,month:jan,tod:11:0,
                                         pod:morning_ger2]])']]

; -  so we have to leave Munich at six o'clock
;    ^ t1003benb7e8r1geySHALLOW
;       (((SUGGEST . 73.986) 8.220) ((INFORM_FEATURE . 78.199) 8.688))
;       [SUGGEST,travelling,has_move:[move,has_source_location:[city,
         has_name='muenchen'],has_departure_time:[date,
         tempex='tempex(en_2920_0,[from:tod:6:0])']]]
```

```
; -  vielleicht fahren wir lieber den Tag davor
;    ^ t1004ageb8e9r1genSHALLOW
;       (((SUGGEST . 65.576) 9.368) ((INFORM . 75.163) 10.737))
;       [SUGGEST,travelling,has_move:[move,has_departure_time:[date,
         tempex='tempex(ge_2920_1,[from:neg_shift(dur(1,days),ana_point)])'
         ]]]
; -  da gibt es einen Zug um zwei Uhr
;      t1004ageb9e10r1geySHALLOW
;       (((SUGGEST . 58.064) 7.258) ((INFORM . 65.774) 8.221))
;       [SUGGEST,travelling,has_move:[move,has_transportation:[rail],
         has_departure_time:[date,tempex='tempex(ge_2920_2,[from:tod:2:0])'
         ]]]

; -  I would prefer to leave at five
;    ^ t1005benb10e11r1genSHALLOW
;       (((SUGGEST . 55.135) 7.876) ((REJECT . 63.196) 9.028))
;       [SUGGEST,travelling,has_move:[move,has_agent:[speaker],
         has_departure_time:[date,tempex='tempex(en_2920_1,[from:tod:5:0])'
         ]]]
; -  I am pretty busy that day
;      t1005benb11e12r1geySHALLOW
;       (((GIVE_REASON . 35.271) 5.878) ((REJECT . 39.514) 6.585))
;       [GIVE_REASON,any_topic,has_agent:[speaker]]

; -  ja gerne k"onnen wir machen
;    ^ t1006ageb12e13r1genSHALLOW
;       (((ACCEPT . 31.367) 6.273) ((SUGGEST . 42.486) 8.497))
;       [ACCEPT,any_topic]
; -  dann brauchen wir noch eine "Ubernachtung
;      t1006ageb13e14r1geySHALLOW
;       (((INIT . 43.727) 7.287) ((SUGGEST . 46.141) 7.690))
;       [INIT,accommodation,has_accommodation:[hotel,has_amount=1]]

; -  yes
;    ^ t1007benb14e15r1geySHALLOW
;       (((ACCEPT . 14.225) 14.225) ((INFORM . 19.302) 19.302))
;       [ACCEPT,any_topic]

; -  ich kenne ein gutes Hotel im Stadtzentrum
;    ^ t1008ageb15e16r1genSHALLOW
;       (((INIT . 71.335) 10.190) ((SUGGEST . 75.288) 10.755))
;       [INIT,accommodation,has_agent:[speaker],has_accommodation:[hotel,
         has_amount=1,has_quality_indication=good,has_location:[nongeo_-
         location,has_name='stadtzentrum',has_loc_spec=in,has_det=def]]]
; -  ein Einzelzimmer kostet achtzig Euro
;      t1008ageb16e17r1geySHALLOW
;       (((INFORM . 57.922) 11.584) ((INFORM_FEATURE . 57.937) 11.587))
;       [INFORM,accommodation,consists_of:[room,has_amount=1,
         has_size=single,has_price:[price,has_amount=80,
         has_currency=euro]]]

; -  that sounds pretty good
;    ^ t1009benb17e18r1genSHALLOW
;       (((ACCEPT . 23.778) 5.944) ((SUGGEST . 40.537) 10.134))
;       [ACCEPT,any_topic]
; -  can you please do the reservations
;      t1009benb18e19r1geySHALLOW
;       (((SUGGEST . 65.944) 10.990) ((REQUEST_SUGGEST . 69.315) 11.552))
;       [SUGGEST,any_topic,has_book_action:[book_action,
         has_agent:[addressee]]]

; -  sicher dann machen wir ein gemeinsames Abendessen in einem Restaurant
;    ^ t1010ageb19e20r1genSHALLOW
;       (((ACCEPT . 87.567) 8.756) ((SUGGEST . 89.498) 8.949))
;       [ACCEPT,entertainment,has_entertainment:[entertainment,
         has_theme:[dine_out],has_location:[nongeo_location,
         has_name='restaurant',has_loc_spec=in,has_det=undef]]]
; -  ich werde einen Tisch reservieren
;      t1010ageb20e21r1geySHALLOW
;       (((COMMIT . 33.705) 6.741) ((REQUEST_COMMIT . 57.783) 11.556))
;       [COMMIT,any_topic,has_book_action:[book_action,has_agent:[speaker]]]

; -  that would be nice
;    ^ t1011benb21e22r1genSHALLOW
;       (((ACCEPT . 30.028) 7.507) ((SUGGEST . 35.490) 8.872))
;       [ACCEPT,any_topic]
; -  let us meet at the station on Wednesday
;      t1011benb22e23r1geySHALLOW
;       (((SUGGEST . 62.319) 7.789) ((ACCEPT . 65.925) 8.240))
;       [SUGGEST,scheduling,has_appointment:[appointment,
         has_location:[nongeo_location,has_name='bahnhof',has_loc_spec=at,
         has_det=def],has_date:[date,
         tempex='tempex(en_2920_2,[from:dow:wed])']]]

; -  um halb zehn am Bahnhof
;    ^ t1012ageb23e24r1geySHALLOW
;       (((ACCEPT . 39.409) 7.881) ((SUGGEST . 42.095) 8.419))
;       [ACCEPT,uncertain_scheduling,has_date:[date,
         tempex='tempex(ge_2920_3,[from:tod:9:30])'],
         has_location:[nongeo_location,has_name='bahnhof']]

; -  good see you then
;    ^ t1013benb24e25r1geySHALLOW
;       (((BYE . 24.403) 6.100) ((ACCEPT . 31.609) 7.902))
;       [BYE,any_topic,has_agent:[addressee]]

; -  bis dann
;    ^ t1014ageb25e26r1geySHALLOW
;       (((BYE . 9.760) 4.880) ((ACCEPT . 22.613) 11.306))
;       [BYE,any_topic]
NIL

======================
DIALOGUE FRAME
======================
begin: 5. July 2002, 11:51 pm
end:   6. August 2002, 2:55 pm
participants:
VM_SPEAKER (SPEAKER2) [en] [b]
VM_SPEAKER (SPEAKER1) [<UNK:SURNAME..Thompson>] [ge] [a]


===========================================
SCHEDULING frame [open]
===========================================
theme:


attitudes: (#<U4: #<PC-VM_SPEAKER SPEAKER1> SUGGEST>)
         ===> ACCEPT
relations: ((MORE_SPECIFIC_THAN . #<PC-APPOINTMENT P2*>))
APPOINTMENT (P5*+0)
  HAS_LOCATION --> CITY (P4*)
                     HAS_NAME=hannover
  HAS_MEETING --> MEETING (P3**)
                    HAS_NAME=geschaeftstreffen
  HAS_DATE --> DATE (P5*)
                 TEMPEX=tempex(ge_2920_0, from:[year:2002,
                                                month:jan,
                                                dom:20,
                                                pod:morning_ger2,
                                                tod:11:0])

attitudes: (#<U21: #<PC-VM_SPEAKER SPEAKER1> SUGGEST>)
         ===> ACCEPT
relations: ((MORE_SPECIFIC_THAN . #<PC-APPOINTMENT P26*>)
            (MORE_SPECIFIC_THAN . #<PC-APPOINTMENT P30***+0>))
APPOINTMENT (P29*+0)
  HAS_LOCATION --> NONGEO_LOCATION (P30***)
                     HAS_NAME=bahnhof
  HAS_DATE --> DATE (P29*)
                 TEMPEX=tempex(ge_2920_3, from:[year:2002,
                                                month:jan,
                                                dom:16,
                                                tod:9:30])
```

```
===========================================
ENTERTAINMENT frame [open]
===========================================
theme:


attitudes: (#<U17: #<PC-VM_SPEAKER SPEAKER1> SUGGEST>)
           ===> ACCEPT
relations: NIL
ENTERTAINMENT (P22**)
  HAS_THEME --> DINE_OUT (P23**)
  HAS_LOCATION --> NONGEO_LOCATION (P24**)
                     HAS_NAME=restaurant

attitudes: (#<U19: #<PC-VM_SPEAKER SPEAKER2> ACCEPT>
            #<U18: #<PC-VM_SPEAKER SPEAKER1> COMMIT>)
           ===> ACCEPT
relations: NIL
BOOK_ACTION (P25*)
  HAS_AGENT --> VM_SPEAKER (SPEAKER1)
                  HAS_LAST_NAME=<UNK:SURNAME..Thompson>
                  HAS_LANGUAGE=ge
                  HAS_CHANNEL=a

===========================================
ACCOMMODATION frame [open]
===========================================
theme:


attitudes: (#<U15: #<PC-VM_SPEAKER SPEAKER2> ACCEPT>
            #<U13: #<PC-VM_SPEAKER SPEAKER1> SUGGEST>)
           ===> ACCEPT
relations: ((MORE_SPECIFIC_THAN . #<PC-HOTEL P16*>))
HOTEL (P17*)
  CONSISTS_OF --> ROOM (P19**)
                    HAS_PRICE --> PRICE (P20**)
                                    HAS_CURRENCY=EURO
                                    HAS_AMOUNT=80
                    HAS_SIZE=SINGLE
                    HAS_AMOUNT=1
  HAS_AMOUNT=1
  HAS_QUALITY_INDICATION=GOOD
  HAS_LOCATION --> NONGEO_LOCATION (P18*)
                     HAS_NAME=stadtzentrum

attitudes: (#<U16: #<PC-VM_SPEAKER SPEAKER2> NIL>)
           ===> ACCEPT
relations: NIL
```

```
BOOK_ACTION (P21*)
  HAS_AGENT --> VM_SPEAKER (SPEAKER1)
                  HAS_LAST_NAME=<UNK:SURNAME..Thompson>
                  HAS_LANGUAGE=ge
                  HAS_CHANNEL=a

===========================================
TRAVELLING frame [open]
===========================================
theme:


attitudes: (#<U10: #<PC-VM_SPEAKER SPEAKER1> ACCEPT>
            #<U8: #<PC-VM_SPEAKER SPEAKER2> SUGGEST>)
         ===> ACCEPT
relations: NIL
JOURNEY (P14*+0)
  HAS_MOVE_THERE --> MOVE (P14*)
                       HAS_TRANSPORTATION --> RAIL (P12**)
                       HAS_DEST_LOCATION --> CITY (P4*)
                                               HAS_NAME=hannover
                       HAS_DEPARTURE_TIME --> DATE (P15*)
                                                TEMPEX=tempex(en_2920_1,
                                                              from:
                                                               [year:2002,
                                                                month:jan,
                                                                dom:19,
                                                                pod:pm,
                                                                tod:5:0])


                   VERBMOBIL ERGEBNISPROTOKOLL Nr. 1
   ----------------------------------------------------------------------

   Teilnehmer: Sprecher B, Thompson
   Datum: 6.8.2002
   Uhrzeit: 15:12 Uhr bis 15:13 Uhr
   Thema: Reise mit Treffen Unterkunft und Freizeitgestaltung
   ----------------------------------------------------------------------

                           GESPR"ACHSERGEBNISSE:



   Terminabsprache:
   Sprecher B und Thompson vereinbarten ein Gesch"aftstreffen am 20.
   Januar 2002 um 11 Uhr am Vormittag in Hannover. Sprecher B und
   Thompson treffen sich am 16. Januar 2002 um halb 10 in einem Bahnhof.
```

```
Reiseplanung:
Eine Reise wurde vereinbart. Die Hinfahrt nach Hannover mit der Bahn
beginnt am 19. Januar 2002 um 5 Uhr am Nachmittag.


Unterkunft:
Ein Hotel in einem Stadtzentrum wurde vereinbart. Ein Einzelzimmer
kostet 80 Euro. Thompson k"ummert sich um die Reservierung.


Freizeit:
Ein Essen in einem Restaurant wurde vereinbart. Thompson k"ummert sich
um die Reservierung.
----------------------------------------------------------------------
               Protokollgenerierung automatisch am 6.8.2002 15:15:58 h
```

A bigger example—the Dialogue of the Week ddw33

This dialogue is the *Dialog der Woche—ddw-33*[1] from the third of November 1999 together with the German and English summary. In the trace, the spoken followed by the recognized and segmented chains for each turn are given. Below the trace, the selected (see section 4.4.1) content of the dialogue memory is given.

```
;*************************************************************
; Dialog: 99_33_AC
; Time  : NIL
;*************************************************************

- hi, I'm calling about the trip to Hanover, in March.
; -  hi and clear
;    ^ t103benb2e3r1genSHALLOW
;       (((GREET . 39.984) 13.328) ((INIT . 44.642) 14.880))
;       [GREET,any_topic]
; -  that trip to Hanover on in on shoot
;      t103benb3e4r1geySHALLOW
;       (((REQUEST_COMMENT . 86.046) 10.755) ((INIT . 88.607) 11.075))
;       [REQUEST_COMMENT,travelling,has_move:[move,has_dest_location:[city,
         has_name='hannover',has_det=unknown]]]

- das is' eine wunderbare Idee, da komm' ich doch glatt mit.
; -  das ist eine wunderbare Idee
```

[1] Dialogue of the week

```
;    ^ t104ageb4e5r1genSHALLOW
;       (((ACCEPT . 28.081) 5.616) ((INFORM . 50.517) 10.103))
;       [ACCEPT,any_topic]
; -  da komme ich doch glaube nicht
;      t104ageb5e6r1geySHALLOW
;       (((GIVE_REASON . 54.244) 9.040) ((ACCEPT . 54.507) 9.084))
;       [GIVE_REASON,any_topic,has_agent:[speaker]]

- shall we travel on the first of March then and return on the second?
; -  <UNK-Female.mi-Sel.Michelle> we travel on the the first of March then
;    ^ t105benb6e7r1genSHALLOW
;       (((SUGGEST . 95.022) 9.502) ((REJECT . 106.574) 10.657))
;       [SUGGEST,travelling,has_move:[move,has_addressee:[person,
         has_sex=fem,has_first_name='<UNK:Female.mi-Sel.Michelle>'],
         has_departure_time:[date,tempex='tempex(en_28042_0,[from:
         [dom:1,month:mar]])']]]
; -  and return on the second
;      t105benb7e8r1geySHALLOW
;       (((SUGGEST . 47.415) 9.483) ((REJECT . 48.738) 9.747))
;       [SUGGEST,travelling,has_move_back:[move,has_departure_time:[date,
         tempex='tempex(en_28042_1,[from:dom:2])']]]

- mir w"urde es am zweiten M"arz besser passen.
; -  mir w"urde es am zweiten M"arz besser passen
;    ^ t106ageb8e9r1geySHALLOW
;       (((SUGGEST . 59.840) 7.480) ((ACCEPT . 68.829) 8.603))
;       [SUGGEST,uncertain_scheduling,has_date:[date,
         tempex='tempex(ge_28042_0,[from:[dom:2,month:mar]])']]

- good so, we we'll leave Hamburg on the first and return on the second //
  I suggest we travel by train.
; -  I would so we were to leave Hamburg on the first
;    ^ t107benb9e10r1genSHALLOW
;       (((INFORM . 96.959) 8.814) ((REJECT . 98.477) 8.952))
;       [INFORM,travelling,has_move:[move,has_source_location:[city,
         has_name='hamburg'],has_departure_time:[date,
         tempex='tempex(en_28042_2,[from:dom:1])']]]
; -  and we can on the second
;      t107benb10e11r1geySHALLOW
;       (((SUGGEST . 42.756) 7.126) ((ACCEPT . 46.975) 7.829))
;       [SUGGEST,uncertain_scheduling,has_date:[date,
         tempex='tempex(en_28042_3,[from:dom:2])']]

- ja, mit dem Zug fahren finde ich gut.
; -  ja mit dem Zug
;    ^ t108ageb11e12r1genSHALLOW
;       (((ACCEPT . 23.735) 5.933) ((SUGGEST . 27.805) 6.951))
;       [ACCEPT,travelling,has_move:[move,has_transportation:[rail]]]
; -  fahren finde ich gut
;      t108ageb12e13r1geySHALLOW
;       (((ACCEPT . 34.604) 8.651) ((SUGGEST . 42.556) 10.639))
;       [ACCEPT,travelling,has_move:[move,has_agent:[speaker]]]

- good, there is a train that leaves at ten in the morning.
  shall we take that train?
; -  there is a train that leaves at ten in the morning so we take that train
;    ^ t109benb13e14r1geySHALLOW
;       (((INFORM_FEATURE . 146.674) 9.167) ((SUGGEST . 151.907) 9.494))
;       [INFORM_FEATURE,travelling,has_move_there:[move,has_transportation:
         [rail],has_arrival_time:[date,
         tempex='tempex(en_28042_4,[from:tod:10:0])'],
         has_arrival_time:[date,tempex='tempex(en_28042_5,
         [from:pod:morning_ger1])'],has_transportation:[rail]]]

- ja, dieser Zeitpunkt passt mir sehr gut.
; -  ja diese Zeitpunkt pa"st mir sehr gut
;    ^ t110ageb14e15r1geySHALLOW
;       (((ACCEPT . 50.438) 7.205) ((SUGGEST . 64.508) 9.215))
;       [ACCEPT,any_topic]

- good, do you know a hotel in Hanover?
; -  good did you know the hotel in Hanover
;    ^ t111benb15e16r1geySHALLOW
;       (((INFORM_FEATURE . 75.444) 9.430) ((SUGGEST . 83.008) 10.376))
;       [INFORM_FEATURE,accommodation,has_agent:[addressee],
         has_accommodation:[hotel,has_location:[city,has_name='hannover',
         has_det=undet],has_det=def]]

- ja, ich kenne mehrere Hotels in Hannover.
; -  ja ich kenne mehrere Hotels in Hannover ja
;    ^ t112ageb16e17r1geySHALLOW
;       (((INFORM . 69.915) 8.739) ((SUGGEST . 74.666) 9.333))
;       [INFORM,accommodation,has_agent:[speaker],has_accommodation:[hotel,
         has_amount_indication=several,has_location:[city,has_name='hannover',
         has_det=undet]]]

- good. can you make reservations for us there?
; -  you make reservations four us there
;    ^ t113benb17e18r1geySHALLOW
;       (((REQUEST_SUGGEST . 72.282) 12.047) ((INFORM_FEATURE . 73.725) 12.287))
;       [REQUEST_SUGGEST,any_topic,has_book_action:[book_action,
         has_agent:[addressee]]]

- k"onn' Sie das bitte wiederholen?
; -  k"onnen Sie das bitte wiederholen
;    ^ t114ageb18e19r1geySHALLOW
;       (((REQUEST . 36.478) 7.295) ((REQUEST_SUGGEST . 50.099) 10.019))
;       [REQUEST,any_topic,has_agent:[addressee]]
```

```
- yes. will you please make reservations at a hotel?
; -  would you please make reservations at the hotel
;    ^ t115benb19e20r1geySHALLOW
;       (((COMMIT . 75.067) 9.383) ((SUGGEST . 83.135) 10.391))
;       [COMMIT,accommodation,has_book_action:[book_action,has_agent:
         [addressee],has_book_theme:[hotel,has_loc_spec=at,has_det=def]]]

- ja, welches Hotel von den dreien m\"ochten wir nehm'?
; -  ja welches Hotel von den dreien m"ochten wir nehmen
;    ^ t116ageb20e21r1geySHALLOW
;       (((INIT . 84.874) 9.430) ((INFORM . 95.995) 10.666))
;       [INIT,accommodation,has_book_action:[book_action,
         has_book_theme:[hotel]]]

- you choose.
; -  yeah she is
;    ^ t117benb21e22r1geySHALLOW
;       (((ACCEPT . 36.332) 12.110) ((INFORM . 41.427) 13.809))
;       [ACCEPT,any_topic]

- ich schlage das Hotel Luisenhof vor.
; -  ich schlage das Hotel lohnt diese denn Hof vor
;    ^ t118ageb22e23r1geySHALLOW
;       (((SUGGEST . 92.379) 10.264) ((COMMIT . 99.272) 11.030))
;       [SUGGEST,accommodation,has_agent:[speaker],
         has_accommodation:[hotel,has_det=def]]

- good. I trust your choice. why don't we meet at the station on Wednesday?
; -  good I will trust your choice
;    ^ t119benb23e24r1genSHALLOW
;       (((ACCEPT . 67.657) 11.276) ((BYE . 69.806) 11.634))
;       [ACCEPT,any_topic,has_agent:[speaker]]
; -  why don't we meet at the station on Wednesday
;      t119benb24e25r1geySHALLOW
;       (((SUGGEST . 67.800) 7.533) ((REQUEST_SUGGEST . 94.176) 10.464))
;       [SUGGEST,scheduling,has_appointment:[appointment,
         has_location:[nongeo_location,has_name='bahnhof',
         has_loc_spec=at,has_det=def],has_date:[date,
         tempex='tempex(en_28042_7,[from:dow:wed])']]]

- ja, dann treffen wir uns doch am Mittwoch am Bahnhof.
; -  ja dann treffen wir uns doch am Mittwoch am Bahnhof
;    ^ t120ageb25e26r1geySHALLOW
;       (((ACCEPT . 60.055) 6.005) ((SUGGEST . 62.715) 6.271))
;       [ACCEPT,scheduling,has_appointment:[appointment,has_date:[date,
         tempex='tempex(ge_28042_1,[from:dow:wed])'],
         has_location:[nongeo_location,has_name='bahnhof',
         has_loc_spec=at,has_det=def]]]
```

```
- good. at nine forty five $A-$M.
; -  great at nine
;    ^ t121benb26e27r1genSHALLOW
;       (((ACCEPT . 29.854) 9.951) ((SUGGEST . 34.135) 11.378))
;       [ACCEPT,uncertain_scheduling,has_date:[date,
         tempex='tempex(en_28042_8,[from:tod:9:0])']]
; -  forty five $A-$M
;      t121benb27e28r1geySHALLOW
;       (((ACCEPT . 47.872) 15.957) ((INFORM_FEATURE . 48.404) 16.134))
;       [ACCEPT,uncertain_scheduling,has_date:[date,
         tempex='tempex(en_28042_9,[from:pod:am])']]

- einverstanden. um Viertel vor zehn treffen wir uns am Bahnhof.
; -  einverstanden um Viertel vor zehn treffen wir uns am Bahnhof
;    ^ t122ageb28e29r1geySHALLOW
;       (((ACCEPT . 74.036) 7.403) ((SUGGEST . 78.333) 7.833))
;       [ACCEPT,scheduling,has_appointment:[appointment,has_date:[date,
         tempex='tempex(ge_28042_2,[from:tod:9:45])'],
         has_location:[nongeo_location,has_name='bahnhof',
         has_loc_spec=at,has_det=def]]]

- good. see you there.
; -  good see you then
;    ^ t123benb29e30r1geySHALLOW
;       (((BYE . 24.033) 6.008) ((ACCEPT . 31.635) 7.908))
;       [BYE,any_topic,has_agent:[addressee]]

- ja, alles klar. wann fahren wir zur"uck?
; -  ja alles klar
;    ^ t124ageb30e31r1genSHALLOW
;       (((ACCEPT . 17.700) 5.900) ((BYE . 20.166) 6.722))
;       [ACCEPT,any_topic]
; -  wann fahren wir zur"uck
;      t124ageb31e32r1geySHALLOW
;       (((REQUEST_SUGGEST . 38.082) 9.520) ((SUGGEST . 43.212) 10.803))
;       [REQUEST_SUGGEST,travelling,has_move_back:[move,
         has_departure_time:[date]]]

- I think, we should return on Thursday evening. there is a six thirty train.
  why don't we take that one?
; -  should return on Thursday evening
;    ^ t125benb32e33r1genSHALLOW
;       (((SUGGEST . 54.203) 10.840) ((ACCEPT . 62.967) 12.593))
;       [SUGGEST,travelling,has_move_back:[move,has_departure_time:[date,
         tempex='tempex(en_28042_10,[from:[dow:thu,pod:evening]])']]]
; -  there is the six thirty
;      t125benb33e34r1genSHALLOW
;       (((INFORM_FEATURE . 44.202) 8.840) ((SUGGEST . 46.394) 9.278))
;       [INFORM_FEATURE,uncertain_scheduling,has_date:[date,
         tempex='tempex(en_28042_11,[from:tod:6:30])']]
; -  train why don't we take that one
;      t125benb34e35r1geySHALLOW
;       (((SUGGEST . 73.156) 10.450) ((REQUEST_SUGGEST . 90.955) 12.993))
;       [SUGGEST,travelling,has_move:[move,has_transportation:[rail]]]

- gut. um Donnerstach Abend fahren wir zur"uck. super Sache.
; -  gut und Donnerstag abend fahren wir zur"uck
;    ^ t126ageb35e36r1genSHALLOW
;       (((SUGGEST . 57.070) 8.152) ((ACCEPT . 64.842) 9.263))
;       [SUGGEST,travelling,has_move_back:[move,has_departure_time:[date,
         tempex='tempex(ge_28042_3,[from:[dow:thu,pod:evening]])']]]
; -  super Sache
;      t126ageb36e37r1geySHALLOW
;       (((ACCEPT . 22.597) 11.298) ((INFORM . 33.639) 16.819))
;       [ACCEPT,any_topic]

- excellent. so, I see you on Wednesday morning. bye.
; -  excellent so I will see you on Wednesday morning
;    ^ t127benb37e38r1genSHALLOW
;       (((ACCEPT . 66.313) 7.368) ((SUGGEST . 71.151) 7.905))
;       [ACCEPT,uncertain_scheduling,has_date:[date,
         tempex='tempex(en_28042_13,[from:[dow:wed,pod:morning_ger1]])']]
; -  bye
;      t127benb38e39r1geySHALLOW
;       (((BYE . 10.547) 10.547) ((ACCEPT . 20.013) 20.013))
;       [BYE,any_topic]

- m"ochten Sie die Preise vom Hotel Luisenhof noch wissen?
; -  m"ochten Sie die Preise vom Hotel Juli diesen Hof noch wissen
;    ^ t128ageb39e40r1geySHALLOW
;       (((REQUEST . 124.347) 11.304) ((SUGGEST . 126.908) 11.537))
;       [REQUEST,accommodation,has_agent:[addressee],has_accommodation:
         [hotel,has_det=def],has_date:[date,
         tempex='tempex(ge_28042_4,[from:month:jul])']]

- no, that's okay. you take care of that.
; -  no that is okay could take care of that mhm
;    ^ t129benb40e41r1geySHALLOW
;       (((REQUEST_SUGGEST . 93.184) 9.318) ((ACCEPT . 98.854) 9.885))
;       [REQUEST_SUGGEST,any_topic]
; -  that
;    ^ t130aenb41e42r1geySHALLOW
;       (((ACCEPT . 13.415) 13.415) ((INFORM . 14.796) 14.796))
;       [ACCEPT,any_topic]

- tsch"uss.
; -  tsch"u"s
;    ^ t131bgeb42e43r1geySHALLOW
;       (((BYE . 10.995) 10.995) ((ACCEPT . 19.939) 19.939))
;       [BYE,any_topic]

- bye.
; -  bye
;    ^ t132aenb43e44r1geySHALLOW
;       (((BYE . 8.090) 8.090) ((ACCEPT . 20.824) 20.824))
;       [BYE,any_topic]
```

Here is the content of the dialogue memory:

```
DIA(119): (dm-show-tm)
======================
DIALOGUE FRAME
======================
begin: 26. September 2002, 10:11 pm
end:   1. October 2002, 1:10 am
participants:
VM_SPEAKER (SPEAKER1) [en] [b]
VM_SPEAKER (SPEAKER2) [FEM] [<UNK:FEMALE.mi-Sel.Michelle>] [ge] [a]


===========================================
ACCOMMODATION frame [open]
===========================================
theme:


attitudes: (#<U17: #<PC-VM_SPEAKER SPEAKER1> NIL>)
           ===> ACCEPT
relations: ((MORE_SPECIFIC_THAN . #<PC-BOOK_ACTION P23*>))
BOOK_ACTION (P24*)
  HAS_AGENT --> VM_SPEAKER (SPEAKER2)
                  HAS_SEX=FEM
                  HAS_FIRST_NAME=<UNK:FEMALE.mi-Sel.Michelle>
                  HAS_LANGUAGE=ge
                  HAS_CHANNEL=a
  HAS_BOOK_THEME --> HOTEL (P25*)

===========================================
TRAVELLING frame [open]
===========================================
theme:


attitudes: (#<U34: #<PC-VM_SPEAKER SPEAKER2> ACCEPT>
            #<U33: #<PC-VM_SPEAKER SPEAKER2> SUGGEST>)
           ===> ACCEPT
```

```
relations: NIL
JOURNEY (P47*+0)
  HAS_MOVE_THERE --> MOVE (P45**)
                       HAS_TRANSPORTATION --> RAIL (P46**)
                       HAS_ARRIVAL_TIME -->
                                      DATE (P18*****)
                                          TEMPEX=tempex(en_28042_5,
                                                        from:[year:2002,
                                                              month:mar,
                                                              dom:2,
                                                              pod:
                                                               morning_ger1])
                       HAS_ADDRESSEE --> VM_SPEAKER (SPEAKER2)
                                           HAS_SEX=FEM
                                           HAS_FIRST_NAME=
                                             <UNK:FEMALE.mi-Sel.Michelle>
                                           HAS_LANGUAGE=ge
                                           HAS_CHANNEL=a
                       HAS_DEPARTURE_TIME -->
                                      DATE (P44***)
                                          TEMPEX=tempex(en_28042_11,
                                                        from:[year:2002,
                                                              month:feb,
                                                              dom:28,
                                                              pod:evening,
                                                              tod:6:30])
  HAS_MOVE_BACK --> MOVE (P47*)
                      HAS_DEPARTURE_TIME --> DATE (P48*)
                                              TEMPEX=
                                                tempex(ge_28042_3,
                                                       from:[year:2002,
                                                             month:feb,
                                                             dom:28,
                                                             dow:thu,
                                                             pod:evening])

===========================================
SCHEDULING frame [open]
===========================================
theme:

attitudes: (#<U26: #<PC-VM_SPEAKER SPEAKER2> SUGGEST>)
           ===> ACCEPT
relations: ((MORE_SPECIFIC_THAN . #<PC-APPOINTMENT P29*>))
APPOINTMENT (P37**)
  HAS_DATE --> DATE (P38**)
                 TEMPEX=tempex(ge_28042_2, from:[year:2002, month:feb,
                                                 dom:27, pod:am, tod:9:45])
  HAS_LOCATION --> NONGEO_LOCATION (P39**)
```

```
HAS_NAME=bahnhof
```

This is the German summary. Please note that Michelle is a confabulation.

```
                VERBMOBIL ERGEBNISPROTOKOLL Nr. 1
----------------------------------------------------------------------

Teilnehmer: Sprecher B, Frau Michelle
Datum: 2.9.2003
Uhrzeit: 15:12 Uhr bis 15:13 Uhr
Thema: Reise mit Treffen und Unterkunft
----------------------------------------------------------------------
                        GESPR"ACHSERGEBNISSE:



Terminabsprache:
Sprecher B und Frau Michelle treffen sich am 26.
Februar 2003 um viertel vor 10 am Morgen am Bahnhof.


Reiseplanung:
Eine Reise wurde vereinbart. Die Hinfahrt findet mit der Bahn statt.
Die Hinfahrt dauert vom 27. Februar 2003 um halb 7 am Abend bis dem 2.
M"arz 2003 am Morgen. Die R"uckreise beginnt am Donnerstag Abend am 27.
Februar 2003.


Unterkunft:
Frau Michelle reserviert ein Hotel.
--------------------------------------------------------------------
               Protokollgenerierung automatisch am 2.9.2003 15:15:58
```

Next, the English summary:

```
                      VERBMOBIL SUMMARY Nr. 1
--------------------------------------------------------------------

Participants: Michelle, Speaker B
Date: 2.9.2003
Time: 15:12 until 15:13 Uhr
Theme: Appointment schedule with trip and accommodation
--------------------------------------------------------------------

                        RESULTS:

Scheduling:
Speaker B and Michelle will meet in the train station on the 26.
of february 2003 at a quarter to 10 in the morning.

Traveling:
A trip was agreed on.  The trip there is by train.  The trip there
lasts from the 27. of february 2003 at six thirty pm until the 2.
of march in the morning.  The trip back starts an thursday the 27.
of february 2003.

Accommodation:
Michelle is taking care of the hotel reservation.

--------------------------------------------------------------------
               Summary generation automatically 2.9.2003 15:15:58
```

The VerbMobil sortal ontology

Sorts						**Examples**
anything	abstract	property				*Angst, Interesse*
		field				*Informatik, Wirtschaft, Sport*
		info_content				*Programm, Plan*
		institution				*BASF, Universität*
		symbol				*Bindestrich, Punkt*
	space_time	temporal	situation	meeting_sit		*treffen, Konferenz, Sitzung*
				communicat_sit		*sagen, danken, Gespräch*
				action_sit		*arbeiten, schreiben, handeln*
				move_sit		*fahren, legen, gehen, Reise*
				position_sit		*liegen, stehen*
				temp_sit		*anfangen, beenden, dauern*
				mental_sit		*passen, denken, Annahme*
			time			*Sommer, Abend, dofw, mofy*
		entity	object	agentive	human	*Frau, Kollege, Leute*
					animal	*Käfer, Hund*
				thing	instrument	*Computer, Telefon*
					info_bearer	*Kalender, Unterlagen*
					food	*Bier, Abendessen*
					vehicle	*Auto, Zug*
				substance		*Luft*
			location	geo_location		*Berlin, Saarbrücken*
				nongeo_location		*Raum, Gebäude*

Figure 4: The sortal ontology in VerbMobil

References

Alexandersson, J., & Becker, T. (2001). Overlay as the Basic Operation for Discourse Processing in a Multimodal Dialogue System. In *Workshop Notes of the IJCAI-01 Workshop on "Knowledge and Reasoning in Practical Dialogue Systems"* (pp. 8–14). Seattle, Washington.

Alexandersson, J., & Becker, T. (2003a). Default Unification for Discourse Modelling. In H. Bunt & R. Muskens (Eds.), *Computing meaning* (Vol. 3). Dordrecht: Kluwer Academic Publishers. (Forthcoming)

Alexandersson, J., & Becker, T. (2003b). The Formal Foundations Underlying Overlay. In *Proceedings of the* 5^{th} *International Workshop on Computational Semantics (IWCS-5).* Tilburg, The Netherlands.

Alexandersson, J., Buschbeck-Wolf, B., Fujinami, T., Kipp, M., Koch, S., Maier, E., Reithinger, N., Schmitz, B., & Siegel, M. (1998). *Dialogue Acts in VERBMOBIL-2 – Second Edition* (Verbmobil-Report No. 226). DFKI Saarbrücken, Universität Stuttgart, Technische Universität Berlin, Universität des Saarlandes.

Alexandersson, J., Engel, R., Kipp, M., Koch, S., Küssner, U., Reithinger, N., & Stede, M. (2000). Modeling Negotiation Dialogues. In W. Wahlster (Ed.), *Verbmobil: Foundations of Speech-to-Speech Translation* (pp. 441–451). Springer-Verlag.

Alexandersson, J., & Poller, P. (1998). Towards Multilingual Protocol Generation For Spontaneous Speech Dialogues. In *Proceedings of the* 9^{th} *International Workhop on Natural Language Generation (INLG-98)* (pp. 198–207). Niagara-On-The-Lake, Ontario, Canada.

Alexandersson, J., & Poller, P. (2000). Generating Multilingual Dialog Summaries and Minutes. In W. Wahlster (Ed.), *Verbmobil: Foundations of Speech-to-Speech Translation* (pp. 509–518). Springer-Verlag.

Alexandersson, J., Poller, P., Kipp, M., & Engel, R. (2000). Multilingual Summary Generation in a Speech–To–Speech Translation System for Multilingual Dialogues. In *Proceedings of the international natural language generation conference (INLG-2000)* (pp. 148–155). Mitzpe Ramon, Israel.

Alexandersson, J., & Reithinger, N. (1997). Learning Dialogue Structures from a Corpus. In *Proceedings of the* 5^{rd} *european conference on speech communication and technology (EUROSPEECH-97)* (pp. 2231–2235). Rhodes.

Alexandersson, J., Reithinger, N., & Maier, E. (1997). Insights into the Dialogue Processing of Verbmobil. In *Proceedings of the* 5^{th} *Conference on Applied Natural Language Processing, ANLP-97* (pp. 33–40). Washington, DC.

Allen, J., & Core, M. (1997, October). *Draft of DAMSL: Dialog Act Markup in Several Layers.*

Allen, J., Schubert, L., Ferguson, G., Heeman, P., Hwang, C. H., Kato, T., Light, M., Martin, N., Miller, B., Poesio, M., & Traum, D. (1995). The TRAINS Project: A case Study In Defining a Conversational Planning Agent. *Journal of Experimental and Theoretical AI*, *7*, 7–48.

Allen, J. F., Miller, B. W., Ringger, E. K., & Sikorski, T. (1996). A robust System for Natural Spoken Dialogue. In A. Joshi & M. Palmer (Eds.), *Proceedings of the* 34^{th} *annual meeting of the association for computational linguistics - (ACL'96)* (pp. 62–70). San Francisco: Morgan Kaufmann Publishers.

Allen, J. F., & Perrault, C. R. (1980). Analyzing Intention in Utterances. *Artificial Intelligence*, *15*, 143–178.

Allwood, J. (1976). *Linguistic Communication as Action and Cooperation - A study In Pragmatics.* Doctoral dissertation, Department of Linguistics, University of Göteborg, University of Göteborg. (Gothenburg Monographs in Linguistics 2)

Allwood, J. (1977). A critical Look at Speech Act Theory. In O. Dahl (Ed.), *Logic, pragmatics and grammars* (pp. 53–99). Lund, Studentlitteratur.

Allwood, J. (1994). Obligations and Options in Dialogue. *Think*, *3*, 9–18.

Allwood, J. (1995a). *An Activity Based Approach to Pragmatics* (Tech. Rep. No. 76). University of Gothenburg Department of Linguistics Box 200, S-405 30 Goeteborg: Linguistics Department, University of Gothenburg. (ISSN 0349-1021)

Allwood, J. (1995b). Dialog as Collective Thinking. In P. Pylkkänen & P. Pylkkö (Eds.), *New directions in cognitive science.* Helsinki. (Also in Pylkkänen, P., Pylkkö, P. and Hautamäki, A. (Eds.) 1977, Brain, Mind and Physics, pp. 205-211, Amsterdam, IOS Press)

Allwood, J. (2000). An Activity Based Approach to Pragmatics. In H. Bunt & B. Black (Eds.), *Abduction, belief and context in dialogue* (pp. 47–80). Amsterdam: John Benjamins.

Allwood, J. (2001). Structure of Dialog. In M. Taylor, D. Bouwhuis, & F. Neal (Eds.), *The structure of multimodal dialogue ii* (pp. 3–24). Amsterdam: John Benjamins.

Allwood, J., Nivre, J., & Ahlsén, E. (1992). On the Semantics and Pragmatics of Linguistic Feedback. *Journal of Semantics*(9), 1–26.

Antoniou, G. (1999). A Tutorial on Default Logics. *ACM Computing Surveys (CSUR) Archive*, *31*(4), 337–359.

Auerswald, M. (2000). Example-Based Machine Translation with Templates. In W. Wahlster (Ed.), *Verbmobil: Foundations of Speech-to-Speech Translation* (pp. 420–429). Springer-Verlag.

Austin, J. (1962). *How To Do Things with Words.* Oxford: Clarendon Press.

Baggia, P., Kellner, A., Perennou, G., Popovici, C., Sturm, J., & Wessel, F. (1999). Language modelling and spoken dialogue systems - the ARISE experience. In *Proceedings of the* 6^{rd} *european conference on speech communication and technology (EUROSPEECH-99)* (pp. 1767–1770). Budapest, Hungary.

Batliner, A., Buckow, J., Niemann, H., Nöth, E., & Warnke, V. (2000). The Prosody Module. In W. Wahlster (Ed.), *Verbmobil: Foundations of Speech-to-Speech Translation* (pp. 106–121). Springer-Verlag.

Becker, T., Finkler, W., Kilger, A., & Poller, P. (1998). An Efficient Kernel for Multilingual Generation in Speech–to–Speech Dialogue Translation. In *Proceedings of the* 36^{th} *annual meeting of the association* *for computational linguistics and the* 17^{th} *international conference on computational linguistics (COLING/ACL-98)* (pp. 110–116). Montreal, Quebec, Canada.

Becker, T., Kilger, A., Lopez, P., & Poller, P. (2000a). An Extended Architecture for Robust Generation. In *Proceedings of the international natural language generation conference (INLG-2000)* (pp. 63–68). Mitzpe Ramon, Israel.

Becker, T., Kilger, A., Lopez, P., & Poller, P. (2000b). The Verbmobil Generation Component VM-GECO. In W. Wahlster (Ed.), *Verbmobil: Foundations of Speech-to-Speech Translation* (pp. 483–498). Springer-Verlag.

Becker, T., & Löckelt, M. (2000). Liliput: a Parameterizable Finite-Domain Constraint Solving Framework and its Evaluation with Natural Language Generation Problems. In *TRICS: Techniques for Implementing Constraint Programming Systems, a CP 2000 workshop* (pp. 101–117). Singapor.

Bilange, E. (1991). A task Independent Oral Dialogue Model. In *Proceedings of the 5th. meeting of the european chapter of the association for computational linguistics – (EACL'91)* (pp. 83–88). Berlin, Germany.

Birkenhauer, C. (1998). *Das Dialoggedächtnis des Übersetzungssystems Verbmobil.* Diplomarbeit, Unversität des Saarlandes.

Block, H. U., Schachtl, S., & Gehrke, M. (2000). Adapting a Large Scale MT System for Spoken Language. In W. Wahlster (Ed.), *Verbmobil: Foundations of Speech-to-Speech Translation* (pp. 396–412). Springer-Verlag.

Borgida, A., Brachman, R. J., McGuinness, D. L., & Resnick, L. A. (1989). CLASSIC: A structural Data Model for Objects. In *Proceedings of the 1989 ACM SIGMOD International Conference on Management of Data* (pp. 59–67). Oregon.

Boros, M., Eckert, W., Gallwitz, F., Görz, G., Hanrieder, G., & Niemann, H. (1996). Towards Understanding Spontaneous Speech: Word Accuracy vs. Concept Accuracy. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP-96)* (Vol. 2, pp. 1009–1012). Philadelphia, PA.

Bos, J., & Heine, J. (2000). Discourse and Dialog Semantics for Translation. In W. Wahlster (Ed.), *Verbmobil: Foundations of Speech-to-Speech Translation* (pp. 337–348). Springer-Verlag.

Bos, J., Schiehlen, M., & Egg., M. (1996). *Definition of the Abstract Semantic Classes for the Verbmobil Forschungsprototyp 1.0* (Verbmobil-Report No. 165). Universität des Saarlandes, IBM Heidelberg, Universität Stuttgart. (ISSN 1434-8845)

Bouma, G. (1990). Defaults in Unification Grammar. In *Proceedings of the 28nd annual meeting of the association for computational linguistics* (pp. 165–172). University of Pittsburgh, Pennsylvania, USA.

Boye, J. (1996). Directional Types in Logic Programming (Doctoral dissertation, Linköping Institute of Technology). *Linköping studies in science and technology.*

Brachman, R., & Schmolze, J. (1985). An Overview of the KL-ONE Knowledge Representation System. *Cognitive Science*, *9*(2), 171–216.

Brown, G. P. (1980). Characterizing Indirect Speech Acts. *American Journal of Computational Linguistics*, *6*(3–4), 150–166.

Bunt, H. (1989). Information Dialogues as Communicative Action in Relation to Partner Modelling and Information Processing. In M. Taylor, F. Néel, & D. Bouwhuis (Eds.), *The Structure of Multimodal Dialogue* (pp. 47–73). Amsterdam: North-Holland Elsevier.

Bunt, H. (1994). Context and Dialogue Control. *Think Quarterly*, *3*(1), 19–31.

Bunt, H. (1995). Dialogue Control Functions and Interaction Design. In R. Beun, M. Baker, & M. Reiner (Eds.), *Dialogue and instruction* (pp. 197–214). Heidelberg: Springer Verlag.

Bunt, H. (2000). Dialogue Pragmatics and Context Specification. In H. C. Bunt & W. J. Black (Eds.), *Abduction, Belief and Context in Dialogue. Studies in Computational Pragmatics* (Vol. 1, pp. 81–150). Amsterdam: John Benjamins Publishing Company.

Bunt, H. C. (1996). Dynamic Interpretation and Dialogue Theory. In M. M. Taylor, F. Néel, & D. G. Bouwhuis (Eds.), *The structure of multimodal dialogue, volume 2.* Amsterdam: John Benjamins.

Burger, S. (1997). *Transliteration spontansprachlicher Daten* (Verbmobil Technisches Dokument No. 56). Universität München.

Buschbeck-Wolf, B. (1997). *Resolution on Demand* (Verbmobil-Report No. 196). IMS, Universitt Stuttgart.

Bußmann, H. (1990). *Lexikon der Sprachwissenschaft* (2 ed.). Stuttgart: Alfred Kröner Verlag.

Cahill, L., Doran, C., Evans, R., Mellish, C., Paiva, D., Reape, M., Scott, D., & Tipper, N. (1999). In Search of a Reference Architecture for NLG Systems. In *Proceedings of the* 7th *european workshop on natural language generation* (pp. 77–85). Toulouse.

Cahill, L., Doran, C., Evans, R., Mellish, C., Paiva, D., Reape, M., Scott, D., & Tipper, N. (2000). Reinterpretation of an Existing NLG System in a Generic Generation Architecture. In *Proceedings of the international natural language generation conference (INLG-2000)* (pp. 69–76). Mitzpe Ramon, Israel.

Carberry, S. (1990). *Plan Recognition in Natural Language Dialogue.* Cambridge, MA: The MIT Press.

Carbonell, J. G., & Goldstein, J. (1998). The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval* (pp. 335–336). Melbourne, Austrailia: ACM.

Carletta, J. (1996). Assessing Agreement on Classification Tasks: The Kappa Statistics. *Computational Linguistics*, *22*(2), 249–254.

Carletta, J., Dahlbäck, N., Reithinger, N., & Walker, M. A. (Eds.). (1997). *Standards for Dialogue Coding in Natural Language Processing.* Schloß Dagstuhl. (Seminar Report 167)

Carletta, J., Isard, A., Isard, S., Kowtko, J. C., Doherty-Sneddon, G., & Anderson, A. H. (1997a). The Reliability of a Dialogue Structure Coding Scheme. *Computational Linguistics*, *1*(23), 13–31.

Carletta, J., Isard, A., Isard, S., Kowtko, J. C., Doherty-Sneddon, G., & Anderson, A. H. (1997b). The Reliability of a Dialogue Structure Coding Scheme. *Computational Linguistics*, *23*(1), 13–32.

Carlson, L. (1983). *Dialogue Games: An Approach to Discourse Analysis* (Vol. 17). Dordrecht, Holland: D. Reidel Publishing Company.

Carpenter, B. (1992). *The Logic of Typed Feature Structures.* Cambridge, England: Cambridge University Press.

Carpenter, B. (1993). Skeptical and Credulous Default Unification with Applications to Templates and Inheritance. In T. Briscoe, V. de Paiva, & A. Copestake (Eds.), *Inheritance, defaults, and the lexicon* (pp. 13–37). Cambridge, CA: Cambridge University Press.

Carroll, J., Copestake, A., Flickinger, D., & Poznanski, V. (1999). An efficient chart generator for (semi-)lexicalist grammars. In *Proceedings of the* 7^{th} *european workshop on natural language generation* (pp. 86–95). Toulouse, France.

Charniak, E., Riesbeck, C. R., McDermott, D. V., & Meehan, J. R. (1987). *Artificial Intelligence Programming* (2 ed.). Hillsdale, NJ: Erlbaum.

Chen, S. F. (1996). *Building Probabalistic Models for Natural Language.* Doctoral dissertation, Harward University Cambridge, Massachusetts.

Chu-Carroll, J. (1998). A statistical Model for Discourse Act Recognition in Dialogue Interactions. In J. Chu-Carroll & N. Green (Eds.), *Working notes from the AAAI spring symposium on applying machine learning to discourse processing* (pp. 12–17). Stanford University.

Chu-Carroll, J., & Brown, M. K. (1997). Tracking Initiative in Collaborative Dialogue Interactions. In *Proceedings of the association for computational linguistics – (ACL'97)* (pp. 262–270). Madrid, Spain.

Chu-Carroll, J., & Brown, M. K. (1998). An Evidential Model for Tracking Initiative in Collaborative Dialogue Interactions. *User Modeling and User-Adapted Interaction*, *8*(3-4), 215–253.

Cohen, J. (1960). A coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, *20*, 37–46.

Cohen, P. R. (1978). *On Knowing What to Say: Planning Speech Acts.* Doctoral dissertation, Department of Computer Science, University of Toronto, Ontario. (Available as Technical Report TR 118)

Copestake, A. (1993). *The Representation of Lexical Semantic Information.* Doctoral dissertation, University of Sussex.

Dahlbäck, N., & Jönsson, A. (1992). An Empirically-Based Computationally Tractable Dialogue Model. In *Proceedings of the* 14^{th} *annual meeting of the cognitive science society* (pp. 785–790). Bloomington, Indiana.

Dahlbäck, N., Jönsson, A., & Ahrenberg, L. (1993). Wizard of Oz Studies – Why and How. *Knowledge-Based Systems*, *6*(4), 258–266.

Dale, R. (1995). *An Introduction to Natural Language Generation* (Tech. Rep.). Microsoft Research Institute (MRI), Macquarie University. (Presented at the 1995 European Summer School on Logic, Language and Information)

Denecke, M. (1999). Integrating Knowledge Sources for the Specification of a Task-Oriented System. In J. Alexandersson, L. Ahrenberg, K. Jokinen, & A. Jönsson (Eds.), *Proceedings of the IJCAI'99 Workshop 'Knowledge and Reasoning in Practical Dialogue Systems'* (pp. 33–40). Stockholm.

Deransart, P., & Małuszyński, J. (1993). *A Grammatical View of Logic Programming.* MIT Press.

Dorna, M. (1996). *The ADT Package for the Verbmobil Interface Term* (Verbmobil-Report No. 104). Universität Stuttgart. (ISSN 1434-8845)

Dorna, M. (2000). *A library Package for the Verbmobil Interface Term* (Verbmobil-Report No. 238). IMS, Universitt Stuttgart. (ISSN 1434-8845)

Early, J. (1970). An Efficient Context-Free Parsing Algorithm. *Communications of the ACM*, *2*(13), 94–102.

Eckert, M., & Strube, M. (2000). Dialogue acts, synchronising units and anaphora resolution. *Journal of Semantics*, *17*(1), 51–89.

Emele, M., Dorna, M., Lüdeling, A., Zinsmeister, H., & Rohrer, C. (2000). Semantic-Based Transfer. In W. Wahlster (Ed.), *Verbmobil: Foundations of Speech-to-Speech Translation* (pp. 359–376). Springer-Verlag.

Endriss, U. (1998). *Semantik zeitlicher Ausdrücke in Terminvereinbarungsdialogen* (Verbmobil-Report No. 227). Technische Universität Berlin.

Feinstein, A., & Cicchetti, D. (1990). High Agreement but Low Kappa: I. The Problems of Two Paradoxes. *Journal of Clinical Epidemiology*, *43*(6), 543–549.

Ferguson, G. M., Allen, J. F., Miller, B. W., & Ringger, E. K. (1996). *The Design and Implementation of the TRAINS-96 System: A Prototype Mixed-Initiative Planning Assistant* (Tech. Rep.). University of Rochester: Computer Science Department. (TRAINS Technical Note 96-5)

Fikes, R. E., & Nilsson, N. J. (1990). STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. In J. Allen, J. Hendler, & A. Tate (Eds.), *Readings in Planning* (pp. 88–97). San Mateo, CA: Kaufmann.

Flickinger, D., Copestake, A., & Sag, I. A. (2000). HPSG Analysis of English. In W. Wahlster (Ed.), *Verbmobil: Foundations of Speech-to-Speech Translation* (pp. 255–264). Springer-Verlag.

Fouvry, F. (2000). Robust Unification for Linguistics. In *1st workshop on RObust Methods in Analysis of Natural language Data - ROMAND'00* (pp. 77–88). Lausanne.

Fouvry, F. (2003). Constraint Relaxation with Weighted Feature Structures. In *International Workshop on Parsing Technologies - IWPT'03* (pp. 103–114). Nancy, France.

Gates, D., Lavie, A., Levin, L., Waibel, A., Gavaldà, M., Mayfield, L., Woszczyna, M., & Zahn, P. (1996). End-to-End Evaluation in JANUS: a Speech-to-Speech Translation System. In E. Maier, M. Mast, & S. LuperFoy (Eds.), *Proceedings of the ECAI-96 Workshop on Dialogue Processing in Spoken Language Systems* (pp. 195–206). Budapest.

Goffman, E. (1981). *Forms of talk.* Oxford: Basil Blackwell.

Gordon, D., & Lakoff, G. (1975). Conversational Postulates. In P. Cole & J. Morgan (Eds.), *Syntax and semantics* (Vol. 3: Speech Acts, pp. 83–106). New York: Academic Press.

Grice, H. P. (1975). Logic and Conversation. In P. Cole & J. Morgan (Eds.), *Syntax and Semantics* (Vol. 3: Speech Acts, pp. 41–58). New York: Academic Press.

Grimes, J. E. (1975). *The Thread of Discourse.* Mouton, The Hague, Paris.

Grosz, B., Joshi, A., & Weinstein, S. (1995). Centering: A Framework for Modeling the Local Coherence of Discourse. *Computational Linguistics*, *2*(21), 203–225.

Grosz, B., & Sidner, C. (1986). Attention, Intentions and the Structure of Discourse. *Computational Linguistics*, *12*(3), 175–204.

Grove, W. M., Andreasen, N. C., McDonald-Scott, P., Keller, M. B., & Shapiro, R. W. (1981). Reliability Studies of Psychiatric Diagnosis, Theory and Practice. *Archives General Psychiatry*, *38*, 408–413.

Grover, C., Brew, C., Manandhar, S., & Moens, M. (1994). Priority Union and Generalization in Discourse Grammars. In *Proceedings of the* 32nd *Annual Meeting of the Association for Computational Linguistics* (pp. 17–24). Las Cruces, New Mexico.

Guinn, C. I. (1996). Mechanisms for Mixed-Initiative Human-Computer Collaborative Discourse. In A. Joshi & M. Palmer (Eds.), *Proceedings of the* 34th *Annual Meeting of the Association for Computational Linguistics - (ACL'96)* (pp. 278–285). San Francisco: Morgan Kaufmann Publishers.

Gwet, K. (2001). *Handbook of Inter-Rater Reliability.* STATAXIS Publishing Company.

Heeman, P., & Strayer, S. (2001). Adaptive Modeling of Dialogue Initiative. In *Proceedings of the NAACL Workshop on Adaption in Dialogue Systems* (pp. 79–80). Pittsburgh, USA.

Heinecke, J., & Worm, K. (1996). *The Verbmobil Semantic Database* (Verbmobil-Report No. 106). Humboldt-Universität Berlin, Universität des Saarlandes. (ISSN 1434-8845)

Heisterkamp, P., & McGlashan, S. (1996). Units of Dialogue Management: An Example. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP-96)* (pp. 200–203). Philadelphia, PA.

Hirschberg, J., & Litman, D. (1987). Now let's Talk About Now; Identifying Cue Phrases Intonationally. In *Proceedings of the* 25th *Conference of the Association for Computational Linguistics – (ACL'87)* (pp. 163–171). Stanford University, Stanford, CA.

Hitzeman, J., Mellish, C., & Oberlander, J. (1997). Dynamic Generation of Museum Web Pages: The Intelligent Labelling Explorer. In *Proceedings of the Museums and the Web Conference* (pp. 107–115). Los Angeles.

Hobbs, J., Appelt, D. E., Bear, J., Israel, D., Kameyama, M., Stickel, M., & Tyson, M. (1996). FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text. In E. Roche & Y. Schabes (Eds.), *Finite State Devices for Natural Language Processing* (pp. 383–406). MIT Press.

Honderich, T. (Ed.). (1995). *The Oxford Companion to Philosophy.* Oxford: Oxford University Press.

Hovy, E., & Marcu, D. (1998). *Automated Text Summarization.* Tutorial at the 36^{th} Annual Meeting of the Association for Computational Linguistics and the 17^{th} International Conference on Computational Linguistics (COLING/ACL-98). Nancy.

Iida, H., & Arita, H. (1992). Natural Language Dialogue Understanding on a Four-layer Plan Recognition Model. *Journal of Information Processing*, *Vol. 15*(No. 1), 60–71.

Imaichi, O., & Matsumoto, Y. (1995). Integration of Syntactic, Semantic and Contextual Information in Processing Grammatically Ill-Formed Inputs. In *Proceedings of the* 14^{th} *International Joint Conference on Artificial Intelligence (IJCAI'95)* (pp. 1435–1440). Montreal, Canada.

Jekat, S., Klein, A., Maier, E., Maleck, I., Mast, M., & Quantz, J. J. (1995). *Dialogue Acts in Verbmobil* (Verbmobil-Report No. 65). Universität Hamburg, DFKI Saarbrücken, Universität Erlangen, TU Berlin.

Jelinek, F. (1990). Self-Organized Language Modeling for Speech Recognition. In A. Waibel & K.-F. Lee (Eds.), *Readings in Speech Recognition* (pp. 450–506). Morgan Kaufmann.

Johnston, M., Bangalore, S., Vasireddy, G., Stent, A., Ehlen, P., Walker, M., Whittaker, S., & Maloor, P. (2002). MATCH: An Architecture for Multimodal Dialogue Systems. In *Proceedings of the* 42^{th} *Annual Meeting of the Association for Computational Linguistics, ACL'02* (pp. 376–383). Philadelphia.

Jönsson, A. (1993). *Dialogue Management for Natural Language Interfaces – an Empirical Approach.* Doctoral dissertation, Linköping Studies in Science and Technology.

Jurafsky, D., Wooters, C., Tajchman, G., Segal, J., Stolcke, A., Fosler, E., & Morgan, N. (1995). Using a Stochastic Context-Free Grammar as a Language Model for Speech Recognition. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP-95* (pp. 189–192). Detroit, MI.

Kameyama, M., Ochitani, R., & Peters, S. (1991a). Resolving Translation Mismatches with Information Flow. In *Proceedings of the* 29th *Annual Meeting of the Association for Computational Linguistics (ACL'91)* (pp. 193–200). Berkley, California.

Kameyama, M., Ochitani, R., & Peters, S. (1991b). Resolving Translation Mismatches with Information Flow. In *Proceedings of the* 29th *Annual Meeting of the Association for Computational Linguistics (ACL'91)* (pp. 193–200). University of California, Berkeley, California.

Kamp, H. (1981). A Theory of Truth and Semantic Representation. In J. Groencndijk, T. M. Janssen, & M. Stokhof (Eds.), *Formal Methods in the Study of Language* (pp. 277–322). Amsterdam: Mathematisch Centrum Tracts.

Kamp, H., & Reyle, U. (1993). *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory.* Dordrecht: Kluwer.

Kamp, H., & Scott, D. R. (1996). Discourse and Dialogue. In R. A. Cole, J. Mariani, H. Uszkoriet, A. Zaenen, & V. Zue (Eds.), *Survey of the State of the Art of Human Language Technology.* Stanford, CA and Giardini, Pisa, Italy: Cambridge University Press. (Available at `http://www.cse.ogi.edu/CSLU/HLTsurvey/`)

Kaplan, R. M. (1987). Three Seductions of Computational Psycholinguistics. In P. Whitelock, H. Somers, P. Bennett, R. Johnson, & M. M. Wood (Eds.), *Linguistic Theory and Computer Applications* (pp. 149–188). London: Academic Press.

Karger, R., & Wahlster, W. (2000). Facts and Figures about the Verbmobil Project. In W. Wahlster (Ed.), *Verbmobil: Foundations of Speech-to-Speech Translation* (pp. 454–467). Springer-Verlag.

Karttunen, L. (1986). D-PATR: A Development Environment for Unification-Based Grammars. In *Proceedings of the* 14th *International Conference on Computational Linguistics (COLING-86)* (pp. 25–29). Bonn, Germany: Institut für angewandte Kommunikations- und Sprachforschung e.V. (IKS).

Karttunen, L. (1998). The Proper Treatment of Optimality in Computational Phonology. In K. Oflazer & L. Karttunen (Eds.), *Finite State Methods in Natural Language Processing* (pp. 1–12). Bilkent University, Ankara, Turkey.

Kasper, W., Bos, J., Schiehlen, M., & Thielen, C. (1999). *Definition of Abstract Semantic Classes* (Verbmobil Technisches Dokument No. 61). DFKI GmbH.

Kautz, H. A. (1987). *A Formal Theory of Plan Recognition.* Doctoral dissertation, Department of Computer Science, University of Rochester. (Also available as Technical Report 215)

Kautz, H. A. (1991). A Formal Theory of Plan Recognition and its Implementation. In J. F. Allen, H. A. Kautz, R. N. Pelavin, & J. D. Tenenberg (Eds.), *Reasoning About Plans* (pp. 69–125). San Mateo (CA), USA: Morgan Kaufmann Publishers.

Kay, M. (1996). Chart Generation. In A. Joshi & M. Palmer (Eds.), *Proceedings of the 34th annual meeting of the association for computational linguistics - (ACL'96)* (pp. 200–204). San Francisco: Morgan Kaufmann Publishers.

Kay, M., Gawron, J. M., & Norvig, P. (1991). *Verbmobil - a Translation System for Face-to-Face Dialog.* CSLI.

Kemp, T., Weber, M., & Waibel, A. (2000). End to End Evaluation of the ISL View4You Broadcast News Transcription System. In *Proceedings of the Conference on Content-Based Multimedia Information Access - RIAO-00.* Paris.

Kipp, M., Alexandersson, J., Engel, R., & Reithinger, N. (2000). Dialog Processing. In W. Wahlster (Ed.), *Verbmobil: Foundations of Speech-to-Speech Translation* (pp. 454–467). Springer-Verlag.

Kipp, M., Alexandersson, J., & Reithinger, N. (1999). Understanding Spontaneous Negotiation Dialogue. In J. Alexandersson, L. Ahrenberg, K. Jokinen, & A. Jönsson (Eds.), *Proceedings of the IJCAI'99 Workshop 'Knowledge and Reasoning in Practical Dialogue Systems'* (pp. 57–64). Stockholm.

Klein, M., Bernsen, N. O., Davies, S., Dybkjr, L., Garrido, J., Kasch, H., Mengel, A., Pirrelli, V., Poesio, M., Quazza, S., & Soria,

C. (1998, July). *Supported Coding Schemes.* MATE Deliverable No. D1.1. DFKI, Saarbrücken, Germany. (Also available from `http://www.dfki.de/mate/d11`)

Klüter, A., Ndiaye, A., & Kirchmann, H. (2000). Verbmobil From a Software Engineering Point of View: System Design and Software Integration. In W. Wahlster (Ed.), *Verbmobil: Foundations of Speech-to-Speech Translation* (pp. 637–660). Springer-Verlag.

Koch, S. (1998). *DLR.* (Unpublished Technical Memo)

Koch, S. (1999). *Analyse und Repräsentation von Dialoginhalten – Bestimmung des zentralen Turngehalts.* Diplomarbeit, Technische Universität Berlin.

Koller, A., & Striegnitz, K. (2002). Generation as dependency parsing. In *Proceedings of the* 40^{th} *Conference of the Association of Computational Linguistics (ACL'02)* (pp. 17–24). Philadelphia.

Kowtko, J. C., Isard, S. D., & Doherty, G. M. (1993). *Conversational Games Within Dialogue* (Tech. Rep. No. HCRC/RP-31). University of Edinburgh: Human Communication Research Centre, Edinburgh Human Communication Research Centre, Glasgow.

Krieger, H.-U. (1995). *TDL*—A Type Description Language for Constraint-Based Grammars. Foundations, Implementation, and Applications. (Doctoral dissertation, Universität des Saarlandes, Department of Computer Science). *Saarbrücken Dissertations in Computational Linguistics and Language Technology.*

Krieger, H.-U. (2001). Greatest Model Semantics for Typed Feature Structures. *Grammars*, *4*(2), 139–165.

Kupiec, J., Pedersen, J. O., & Chen, F. (1995). A Trainable Document Summarizer. In *Proceedings of the* 18^{th} *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 68–73). Tampere, Finland.

Küssner, U., & Stede, M. (1995). *Zeitliche Ausdrücke: Repräsentation und Inferenz* (Tech. Rep. No. Verbmobil Memo 100). Technische Universität Berlin. (In German)

Lakoff, R. (1973). Logic of Politeness or Minding Your P's and Q's. In C. Colum et al (Ed.), *Papers from the* 9^{th} *Regional Meeting of Chicago Linguistic Society* (pp. 292–305). Chicago.

Lambert, L. (1993). *Recognizing Complex Discourse Acts: A Tripartite Plan-Based Model of Dialogue.* Doctoral dissertation, University of Delaware, Newark, Delaware.

Larsson, S. (1998a). *Coding Schemas for Dialogue Moves* (Tech. Rep.). Department of Linguistics, Göteborg University. (Technical Report from the S-DIME Project)

Larsson, S. (1998b). Using a Type Hierarchy to Characterize Reliability of Coding Schemas for Dialogue Moves. In *Proceedings of* 2^{nd} *International Conference on Cooperative Multimodal Communication, Theory and Applications (CMC/98).* Tilburg University, Tilburg.

Lascarides, A., & Copestake, A. A. (1999). Default Representation in Constraint-Based Frameworks. *Computational Linguistics*, *25*(1), 55–105.

Levin, J. A., & Moore, J. A. (1977). Dialogue Games: Metacommunication Structures for Natural Language Interaction. *Cognitive Science 1(4)*, *1(4)*, 395–420.

Levin, L., Bartlog, B., Llitjos, A. F., Gates, D., Lavie, A., Wallace, D., Watanabe, T., & Woszczyna, M. (2000). Lessons Learned from a Task-Based Evaluation of Speech-to-Speech Machine Translation. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC-2000)* (pp. 721–724). Athens, Greece.

Levin, L., Gates, D., Lavie, A., & Waibel, A. (1998). An Interlingua Based on Domain Actions for Machine Translation of Task-Oriented Dialogues. In *Processings of the International Conference on Spoken Language Processing (ICSLP-98).* Sydney Convention and Exibition Centre, Darling Harbour, Sydney.

Levin, L., Glickman, O., Qu, Y., Gates, D., Lavie, A., Rose, C., Ess-Dykema, C. V., & Waibel, A. (1995). Using Context in Machine Translation of Spoken Language. In *Proceedings of Theoretical and Methodological Issues in Machine Translation (TMI-95)* (pp. 173–187). Leuven, Belgium.

Levinson, S. C. (1983). *Pragmatics.* Cambridge University Press.

Linell, P., & Gustavsson, L. (1987). *INITIATIV OCH RESPONS Om dialogens dynamik, dominans och koherens* (Vol. SIC 15). University of Linköping, Studies in Communication.

Litman, D. J., & Allen, J. A. (1987). A Plan Recognition Model for Subdialogue in Conversations. *Cognitive Science*, *11*, 163–200.

Löeckelt, M., Becker, T., Pfleger, N., & Alexandersson, J. (2002). Making Sense of Partial. In Bos, Foster, & Matheson (Eds.), *Proceedings of the* 6*th* *Workshop on the Semantics and Pragmatics of Dialogue (EDILOG 2002)* (pp. 101–107). Edinburgh.

Luperfoy, S. (1992). The Representation of Multimodal User Interface Dialogues Using Discourse Pegs. In *Proceedings of acl'92* (pp. 22–31). Deleware, USA.

Maier, E. (1996). Context Construction as Subtask of Dialogue Processing – The Verbmobil Case. In A. Nijholt, H. Bunt, S. LuperFoy, G. V. van Zanten, & J. Schaake (Eds.), *Proceedings of the* 11*th* *Twente Workshop on Language Technology, TWLT, Dialogue Management in Natural Language Systems* (pp. 113–122). Enschede, Netherlands.

Maier, E. (1997). *Evaluating a Scheme for Dialogue Annotation* (Verbmobil-Report No. 193). DFKI GmbH.

Malenke, M., Bäumler, M., & Paulus, E. (2000). Speech Recognition Performance Assessment. In W. Wahlster (Ed.), *Verbmobil: Foundations of Speech-to-Speech Translation* (pp. 583–591). Springer-Verlag.

Mani, I. (2000-2001). Summarization Evaluation: An Overview. In *Proceedings of the* 2*nd* *NTCIR Workshop on Research in Chainese & Japanese Text Retrieval and Text Summarization.* Tokyo, Japan. (ISBN: 4-924600-96-2)

Mani, I., House, D., Klein, G., Hirschman, L., Obrist, L., Firmin, T., Chrzanowski, M., & Sundheim, B. (1998). *The Tipster SUMMAC Text Summarization Evaluation - final Report* (Tech. Rep.). The MITRE Corp.

Mani, I., & Maybury, M. (Eds.). (1999). *Advances in Automatic Text Summarization.* Cambridge, MA: MIT Press.

Mann, W. C., & Thompson, S. A. (1988). Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. *Text*, *3*(8). (Also available as USC Information Sciences Institute Research Report RR-87-190)

Marcu, D. (1997). From Local to Global Coherence: A Bottom-Up Approach to Text Planning. In D. Traum (Ed.), *AAAI/IAAI* (pp. 629–635). Menlo Park, California: American Association for Artificial Intelligence.

Marcu, D. (1998). Improving Summarization Through Rhetorical Parsing Tuning. In *The* 6*th* *workshop on very large corpora* (pp. 206–215). Montreal, Canada.

Marcu, D. (1999). Discourse Trees are Good Indicators of Importance in Text. In I. Mani & M. Maybury (Eds.), *Advances in Automatic Text Summarization* (pp. 123–136). Cambridge, MA: MIT Press.

McGlashan, S., Fraser, M., Nigel, G., Bilange, E., Heisterkamp, P., & Youd, N. J. (1992). Dialogue Management for Telephone Information Systems. In *Proceedings of the* 3*rd* *Conference in Applied Natural Language Processing, ANLP-92* (pp. 245–246). Trento, Italy.

McKeown, K. R. (1985). *Text Generation - Using Discourse Strategies and Focus Constraints to Generate Natural Language Text.* Cambridge, GB: Cambridge University Press.

McKeown, K. R., & Radev, D. R. (1995). Generating Summaries of Multiple News Articles. In *Proceedings,* 18*th* *annual international ACM SIGIR conference on research and development in information retrieval* (pp. 74–82). Seattle, Washington.

Mellish, C., O'Donnell, M., Oberlander, J., & Knott, A. (1998). An Architecture for Opportunistic Text Generation. In E. Hovy (Ed.), *Proceedings of the* 9*th* *International Workshop on Natural Language Generation (INLG-98)* (pp. 28–37). New Brunswick, New Jersey: Association for Computational Linguistics.

Mey, J. L. (2001). *Pragmatics: An Introduction* (2 ed.). Blackwell Publishers. (ISBN 0-631-21132-2)

Moore, D. C. (2002). *The IDIAP Smart Meeting Room* (Tech. Rep. No. IDIAP-Com 02-07). IDIAP.

Moore, J. D., & Paris, C. (1993). Planning Text for Advisory Dialogues: Capturing Intentional and Rhethorical Information. *Computational Linguistics*, *19*, 652–694.

Morgan, N., Baron, D., Edwards, J., Ellis, D., Gelbart, D., Janin, A., Pfau, T., Shriberg, E., & Stolcke, A. (2002). The Meeting Project at ICSI. In *The First International Human Language Technology Conference on Human Language Technology Research HLT-2002* (pp. 246–252). San Diego, California, USA.

Morris, C. (1938). Foundations of the Theory of Signs. In Morris (Ed.), *Writings on the general theory of signs.* The Hague, Mouton. (1971)

Ninomiya, T., Miyao, Y., & Tsujii, J. (2002). Lenient Default Unification for Robust Processing within Unification Based Grammar Formalisms. In *Proceedings of the* 19*th* *international conference on computational linguistics (COLING-2002)* (pp. 744–750). Taipei, Taiwan.

Novick, D. G. (1988). *Control of Mixed-Initiative Discourse Through Meta-Locutionary Acts: A Computational Model.* Doctoral dissertation, University of Oregon.

O'Donnell, M., Knott, A., Oberlander, J., & Mellish, C. (2000). Optimising Text Quality in Generation from Relational Databases. In *Proceedings of the international natural language generation conference (INLG-2000)* (pp. 133–140). Mitzpe Ramon, Israel.

Pfleger, N. (2002). *Discourse Processing for Multimodal dialogues and its Application in Smartkom.* Diplomarbeit, Unversität des Saarlandes.

Pfleger, N., Alexandersson, J., & Becker, T. (2002). Scoring Functions for Overlay and their Application in Discourse Processing. In *Proceedings of KONVENS 2002* (pp. 139–146). Saarbrücken.

Pfleger, N., Alexandersson, J., & Becker, T. (2003). A Robust and Generic Discourse Model for Multimodal Dialogue. In *Workshop Notes of the IJCAI-03 Workshop on "Knowledge and Reasoning in Practical Dialogue Systems"* (pp. 64–70). Acapulco, Mexico.

Pfleger, N., Engel, R., & Alexandersson, J. (2003). Robust Multimodal Discourse Processing. In Kruijff-Korbayova & Kosny (Eds.), *Proceedings of Diabruck:* 7*th* *Workshop on the Semantics and Pragmatics of Dialogue* (pp. 107–114). Wallerfangen, Germany.

Pinkal, M., Rupp, C., & Worm, K. (2000). Robust Semantic Processing of Spoken Language. In W. Wahlster (Ed.), *Verbmobil: Foundations of Speech-to-Speech Translation* (pp. 322–336). Springer-Verlag.

Poesio, M., & Mikheev, A. (1998). The Predictive Power of Game Structure in Dialogue Act Recognition: Experimental Results using Maximum Entropy Estimation. In *Proceedings of the international conference on spoken language processing (ICSLP-98)* (pp. 2235–2238). Sydney, Australia.

Poesio, M., & Traum, D. R. (1997). Conversational Actions and Discourse Situations. *Computational Intelligence*, *13*(3), 309–347.

Pollack, M. E., Hirschberg, J., & Webber, B. L. (1982). User Participation in the Reasoning Processes of Expert Systems. In *Proceedings of the 2nd national conference on artificial intelligence (AAAI-82)* (pp. 358–356). Pittsburgh, PA.

Prüst, H. (1992). *On Discourse Structuring, VP Anaphora and Gapping.* Doctoral dissertation, Universiteit van Amsterdam.

Qu, Y., Di Eugenio, B., Lavie, A., Levin, L. S., & Rose, C. P. (1997). Minimizing Cumulative Error in Discourse Context. In E. Maier, M. Mast, & S. LuperFoy (Eds.), *ECAI workshop on dialogue processing in spoken language systems* (Vol. 1236, pp. 171–182). Springer.

Qu, Y., Rose, C. P., & Di Eugenio, B. (1996). Using Discourse Predictions for Ambiguity Resolution. In *Proceedings of the* 16^{th} *international conference on computational linguistics (COLING-96)* (pp. 358–363). Copenhagen.

Rambow, O., Bangalore, S., & Walker, M. (2001). Natural Language Generation in Dialog Systems. In *Proceedings of the* 1^{st} *international conference on human language technology research - (hlt'2001).* San Diego, USA.

Reber, A. S. (1996). *The Penguin Dictionary of Psychology.* Putnam Inc.

Reiter, E. (1994). Has a Consensus NL Generation Architecture Appeared, and is it Psycholinguistically Plausible? In 7^{th} *international workshop on natural language generation* (pp. 163–170). Kennebunkport, Maine.

Reiter, E., & Dale, R. (2000). *Building Natural-Language Generation Systems.* Cambridge University Press.

Reithinger, N. (1995). Some Experiments in Speech Act Prediction. In *Working Notes from the AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation* (pp. 126–131). University of Stanford.

Reithinger, N. (1999). Robust Information Extraction in a Speech Translation System. In *Proceedings of the* 6^{rd} *european conference on speech communication and technology (EUROSPEECH-99)* (pp. 2427–2430). Budapest, Hungary.

Reithinger, N. (2000). *Leave One Out Experiments for Statistical Dialogue Act Recognition* (Verbmobil-Memo No. 146). DFKI Saarbrücken.

Reithinger, N., Alexandersson, J., Becker, T., Blocher, A., Engel, R., Löeckelt, M., Müeller, J., Pfleger, N., Poller, P., Streit, M., & Tschernomas, V. (2003). Smartkom - Adaptive and Flexible Multimodal Access to Multiple Applications. In *Proceedings of ICMI 2003.* Vancouver, B.C. (forthcoming)

Reithinger, N., & Engel, R. (2000). Robust Content Extraction for Translation and Dialog. In W. Wahlster (Ed.), *Verbmobil: Foundations of Speech-to-Speech Translation* (pp. 428–440). Springer-Verlag.

Reithinger, N., Engel, R., Kipp, M., & Klesen, M. (1996). Predicting Dialogue Acts for a Speech-To-Speech Translation System. In *Proceedings of the International Conference on Speech and Language Processing (ICSLP-96)* (pp. 654–657). Philadelphia, PA.

Reithinger, N., & Kipp, M. (1998). Large Scale Dialogue Annotation in Verbmobil. In *Proceedings of the* 10^{th} *european summer school in logic, language and information (ESSLLI-98) workshop on recent advances in corpus annotation.* Saarbrücken, Germany.

Reithinger, N., Kipp, M., Engel, R., & Alexandersson, J. (2000). Summarizing Multilingual Spoken Negotiation Dialogues. In *Proceedings of the* 38^{th} *conference of the association for computational linguistics (ACL'2000)* (pp. 310–317). Hong Kong, China.

Reithinger, N., & Klesen, M. (1997). Dialogue Act Classification Using Language Models. In *Proceedings of the* 5^{rd} *european conference on speech communication and technology (EUROSPEECH-97)* (pp. 2235–2238). Rhodes.

Rupp, C., Spilker, J., Klarner, M., & Worm, K. L. (2000). Combining Analyses from Various Parsers. In W. Wahlster (Ed.), *Verbmobil: Foundations of Speech-to-Speech Translation* (pp. 311–321). Springer-Verlag.

Sacks, H., Schegloff, E. A., & Jefferson, G. (1974). A Simplest Systematics for the Organisation of Turn-Taking in Conversation. *Language*, *50*, 696–735.

Scha, R., & Polanyi, L. (1988). An Augmented Context Free Grammar for Discourse. In *Proceedings of the* 12^{th} *international conference on computational linguistics (COLING-88)* (pp. 573–577). Budapest.

Scheffler, T. (2003). *Generation With TAG – A Semantics Interface and Syntactic Realizer.* Diplomarbeit, Unversität des Saarlandes.

Schegloff, E. A., & Sacks, H. (1973). Opening up Closings. *Semiotica*, *4*(VIII), 289–327.

Schiehlen, M. (2000). Semantic Construction. In W. Wahlster (Ed.), *Verbmobil: Foundations of Speech-to-Speech Translation* (pp. 200–216). Springer-Verlag.

Schmitz, B., & Quantz, J. (1995). Dialogue Acts in Automatic Dialogue Interpreting. In *Proceedings of the* 6^{th} *international conference on theoretical and methodological issues in machine translation (TMI-95)* (pp. 33–47). Leuven, Belgium.

Searle, J. R. (1969). *Speech Acts.* Cambridge, GB: University Press.

Searle, J. R. (1975). Indirect Speech Acts. In P. Cole & J. Morgan (Eds.), *Syntax and semantics* (Vol. 3: Speech Acts). New York: Academic Press.

Severinson-Eklund, K. (1983). *The Notion of Language Game: A Natural Unit of Dialogue and Discourse* (Tech. Rep. No. SIC 5). University of Linköping, Department of Communication studies.

Sidner, C. L. (1994). An Artificial Discourse Language for Collaborative Negotiation. In B. Hayes-Roth & R. Korf (Eds.), *Proceedings of the* 12^{th} *national conference on artificial intelligence (AAAI-94)* (pp. 814–819). Menlo Park, California: AAAI Press.

Siegel, M. (1996). *Die maschinelle übersetzung aufgabenorientierter Japanisch-Deutscher Dialoge. Lösungen für Translation Mismatches.* Doctoral dissertation, Universität Bielefeld.

Siegel, M. (2000). HPSG Analysis of Japanese. In W. Wahlster (Ed.), *Verbmobil: Foundations of Speech-to-Speech Translation* (pp. 265–280). Springer-Verlag.

Simpson, A., & Fraser, N. M. (1993). Black Box and Glass Box Evaluation of the SUNDIAL System. In *Proceedings of the* 3^{rd} *european conference on speech communication and technology (EUROSPEECH-93)* (pp. 1423–1426). Berlin, Germany.

Sinclair, J. M., & Coulthard, M. (1975). *Towards an Analysis of Discourse: The English used by Teachers and Pupils.* London: Oxford University Press.

Siskind, J. M., & McAllester, D. A. (1993). Nondeterministic Lisp as a Substrate for Constraint Logic Programming. In R. Fikes & W. Lehnert (Eds.), *Proceedings of the* 11^{th} *National Conference on Artificial Intelligence (AAAI-93)* (pp. 133–138). Menlo Park, California: AAAI Press.

Smith, R. W. (1997). Practical Issues in Mixed-Initiative Natural Language Dialog: An Experimental Perspective. In *Proceedings of the AAAI spring symposium on computational models for mixed initiative interaction* (pp. 158–162). Stanford University.

Stede, M. (1999). *Lexical Semantics and Knowledge Representation in Multilingual Text Generation.* Boston/Dordrecht/London: Kluwer Academic Publishers.

Steele, G. L. (1984). *Common LISP.* Digital Press.

Stolcke, A. (1994a). *Bayesian Learning of Probabilistic Language Models.* Doctoral dissertation, University of California at Berkeley.

Stolcke, A. (1994b). *How to Boogie: A Manual for Bayesian Object-oriented Grammar Induction and Estimation.* International Computer Sience Institute of Berkley, California.

Stolcke, A. (1995). An Efficient Probabilistic Context-Free Parsing Algorithm that Computes Prefix Probabilities. *Computational Linguistics*, *21*, 165–201.

Strayer, S., & Heeman, P. (2001). Reconciling Initiative and Discourse Structure. In *Proceedings of the 2nd SIGdial Workshop on Discourse and Dialogue* (pp. 153–161). Aalborg, Denmark.

Tessiore, L., & Hahn, W. von. (2000). Functional Validation of a Machine Interpretation System: Verbmobil. In W. Wahlster (Ed.), *Verbmobil: Foundations of Speech-to-Speech Translation* (pp. 613–636). Springer-Verlag.

Teufel, S., & Moens, M. (2000). What's yours and what's mine: Determining Intellectual Attribution in Scientific Text. In *SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC'00)* (pp. 110–117). Hong Kong, China.

Teufel, S., & Moens, M. (2002). Summarizing Scientific Articles: Experiments with Relevance and Rhetorical Status. *Computational Linguistics*, *28*(4), 409–445. (Special issue on Summarization)

Thompson, W., & Walter, S. (1988). A Reappraisal of the Kappa Coefficient. *Journal of Clinical Epidemiology*, *41*(10), 949–958.

Touretzky, D. (1986). *The Mathematics of Inheritance Systems.* Morgan Kaufmann.

Tsang, E. (1993). *Foundations of constraint satisfaction.* London: Academic Press Ltd.

Uebersax, J. (1987). Diversity of Decision-Making Models and the Measurement of Interrater Agreement. *Psychological Bulletin*, *101*, 140–146.

Uebersax, J. S. (1988). Validity Inferences from Interobserver Agreement. *Psychological Bulletin*, *104*, 405–416.

Vauquois, B. (1975). La Traduction Automatique à Grenoble. *Documents de Linguistique Quantitative*, *24*. (Paris: Dunod)

Vilain, M. B. (1990). Getting Serious about Parsing Plans: a Grammatical Analysis of Plan Recognition. In *Proceedings of American Association for Artificial Intelligence* (pp. 190–197). Boston, MA.

Vogel, S., Och, F. J., Tillmann, C., Nieen, S., Sawaf, H., & Ney, H. (2000). Statistical Methods for Machine Translation. In W. Wahlster (Ed.), *Verbmobil: Foundations of Speech-to-Speech Translation* (pp. 379–395). Springer-Verlag.

Wahlster, W. (Ed.). (2000). *Verbmobil: Foundations of Speech-to-Speech Translation.* Springer-Verlag.

Wahlster, W. (2001). Robust Translation of Spontaneous Speech: A multi-Engine Approach. In *Proceedings of the* 17th *international joint conference on artificial intelligence (IJCAI'01)* (Vol. 2, pp. 1484–1493). Seattle, Washington: San Francisco: Morgan Kaufmann.

Wahlster, W. (2003). Towards Symmetric Multimodality: Fusion and Fission of Speech, Gesture, and Facial Expression. In A. Günter, R. Kruse, & B. Neumann (Eds.), *KI 2003: Advances in Artificial Intelligence. Proceedings of the* 26th *German Conference on Artificial Intelligence* (pp. 1–18). Hamburg: Berlin, Heidelberg: Springer.

Wahlster, W., Reithinger, N., & Blocher, A. (2001). Smartkom: Multimodal Communication with a Life-Like Character. In *Proceedings of the* 7rd *European Conference on Speech Communication and Technology (EUROSPEECH 2001 - Scandinavia)* (pp. 2231–2235). Aalborg, Denmark.

Waibel, A., Bett, M., Metze, F., Ries, K., Schaaf, T., Schultz, T., Soltau, H., Yu, H., & Zechner, K. (2001). Advances in Automatic Meeting Record Creation and Access. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP-2001.* Salt Lake City, UT.

Waibel, A., Soltau, H., Schultz, T., Schaaf, T., & Metze, F. (2000). Multilingual Speech Recognition. In W. Wahlster (Ed.), *Verbmobil: Foundations of Speech-to-Speech Translation* (pp. 33–45). Springer-Verlag.

Walker, M., & Whittaker, S. (1990). Mixed Initiative in Dialogue: An Investigation into Discourse Segmentation. In *Proceedings of the* 28th *Conference of the Association for Computational Linguistics (ACL'90)* (pp. 70–78). University of Pittsburgh.

Ward, N. (2000). Issues in the Transcription of English Conversational Grunts. In *Proceedings of the acl 2000 workshop* 1st sigdial workshop on discourse and dialogue. Hong Kong.

Weizenbaum, J. (1966). ELIZA — A Computer Program for the Study of Natural Language Communication between Man and Machine. *Communications of the ACM*, *9*(1), 36–44.

Whittaker, S., & Stenton, P. (1988). Cues and Control in Expert-Client Dialogues. In *Proceedings of the* 26^{th} *annual meeting of the association for computational linguistics (ACL'88)* (pp. 123–130). Buffalo, NY.

Winograd, T. (1972). *Understanding Natural Language.* New York: Academic Press. (Also published in Cognitive Psychology , 3:1 (1972), pp. 1–191.)

Wittgenstein, L. (1974). *Philosophical Grammar.* Oxford: Basil Blackwell.

Woods, W. (1981). Procedural Semantics as a Theory of Meaning. In A. Joshi, B. L. Webber, & I. Sag (Eds.), *Elements of Discourse Understanding* (pp. 300–334). Cambridge: Cambridge University Press.

Woods, W., & Kaplan, R. (1977). Lunar Rocks in Natural English: Explorations in Natural Language Question Answering. *Fundamental Studies in Computer Science*, *Linguistic Structures Processing*(5), 521–569.

Young, S. (1996). Large Vocabulary Continuous Speech Recognition. *IEEE Signal Processing Magazine*, *5*(13), 45–57.

Zechner, K. (2001a). Automatic Generation of Concise Summaries of Spoken Dialogues in Unrestricted Domains. In *Proceedings of the* 24^{th} *ACM SIGIR International Conference on Research and Development in Information Retrieval* (pp. 199–207). New Orleans, LA.

Zechner, K. (2001b). *Automatic Summarization of Spoken Dialogues in Unrestricted Domains.* Doctoral dissertation, Carnegie Mellon University, School of Computer Science, Language Technologies Institute. (Also printed as: Technical Report CMU-LTI-01-168.)

Zechner, K. (2002). Automatic Summarization of Open-Domain Multiparty Dialogues in Diverse Genres. *Computational Linguistics*, *28*(4), 447–485. (Special issue on Summarization)