SPEECH AND HUMAN-MACHINE DIALOG

Wolfgang Minker

Samir Bennacef

KLUWER ACADEMIC PUBLISHERS

SPEECH AND HUMAN-MACHINE DIALOG

THE KLUWER INTERNATIONAL SERIES IN ENGINEERING AND COMPUTER SCIENCE

SPEECH AND HUMAN-MACHINE DIALOG

by

Wolfgang Minker
University of Ulm
Germany

Samir Bennacef
Vecsys
France

KLUWER ACADEMIC PUBLISHERS
NEW YORK, BOSTON, DORDRECHT, LONDON, MOSCOW

eBook ISBN: 1-4020-8037-9
Print ISBN: 1-4020-8036-0

Created in the United States of America

Visit Springer's eBookstore at: http://www.ebooks.kluweronline.com
and the Springer Global Website Online at: http://www.springeronline.com

Contents

Preface vii

1. INTRODUCTION 1
   1 Language 2
   2 Dialog and Computer 3
   3 Human-Machine Spoken Language Dialog 4
   4 Spoken Language Dialog System 10
   5 Projects and Research Applications 10

2. ROBUST SPOKEN LANGUAGE UNDERSTANDING 13
   1 Introduction 13
   2 Case Grammar Formalism 14
   3 Case Grammar in the LIMSI-L'ATIS System 15
   4 Conclusion 21

3. SPOKEN LANGUAGE DIALOG MODELING 23
   1 Introduction 23
   2 Task Modeling 23
   3 Human-Machine Spoken Language Dialog Modeling 29
   4 Dialog System Example 73
   5 Conclusion 79

4. CONCLUSION 81

References 85

Index 91

This page intentionally left blank

Preface

The conjunction of several factors having occurred throughout the past few years will make humans significantly change their behavior vis-à-vis machines. In particular the use of speech technologies will become normal in the professional domain, but also in everyday life. The performance of speech recognition components has significantly improved: only within ten years we have passed from systems able to recognize isolated words uttered by a single speaker using a limited lexicon of around 50 words to systems able to recognize continuous speech with an unlimited vocabulary uttered by any speaker; or to systems able to carry a spontaneous dialog with a vocabulary of a few thousands of words over the telephone, on a well-defined topic (e.g., information on train or airplane schedules). The development of microelectronics, considered to be at the origin of these significant results, also favors hardware miniaturization enabling the integration of particularly time and memory consuming signal processing algorithms into standard processors and portable systems. Finally, the expanding computer and telephone networks allow the general public an immediate location-independent access to large databases.

Under these circumstances, the state-of-the-art of research and achievements in the domain of human-machine spoken language dialog systems, proposed by Wolfgang Minker and Samir Bennacef, seems particularly relevant. In fact, speech enables access to various information services from a simple standard telephone: the number of calls (up to one and a half million) received per month by systems developed at Philips, CSELT, Nuance, etc., shows the adequacy of these vocal servers to the user needs. Speech, combined with a touch screen, also improves the communication with information terminals, rendering them more natural, effective and fast.

However, the use of human-machine spoken language dialog systems is still not generalized. This may be due to the fact that for a long time, a low word error rate has been considered sufficient for integrating a speech recognition component into an end-to-end human-machine interface. The importance of higher language levels including understanding and dialog have been neglected. In fact, while significant progress has been made in speech recognition, the results achieved by the syntaxico-semantic and pragmatic analysis components and by the modeling of corresponding knowledge were not likewise convincing. It was generally assumed that artificial intelligence techniques only validated on text data could directly be applied to spoken language processing. This, however, turned to ignore the specificity of spoken language, in particular its high variability, and the variety of the required knowledge.

In this perspective, the book of Wolfgang Minker and Samir Bennacef represents a remarkable progress for at least two reasons. Firstly, because it places a focus on the need for a correct modelization of the higher (semantic and pragmatic) speech levels; secondly, because it describes in detail a very interesting process of dialog modeling, which has been integrated in several prototype demonstrators.

With respect to the first issue, the work underlines the importance of a semantic representation well adapted to the processing of spoken language utterances, that does not always respect the structure of the written language. One suggested formalism has been an extension of the case grammars that allows the system to cope with the dislocated and asyntactic character of speech. The choice of the semantic cases is illustrated by a description of the air travel information system L'ATIS, a French version of the American ATIS (Air Travel Information Services) application introduced by DARPA (Defense Research Projects Agency) for the test campaigns of spoken language dialog systems.

With respect to the second issue, after a presentation of the different state-of-the-art dialog model types (structural, plan-directed, logical, task-directed), the book exhaustively describes a dialog model that is related to the structural models having their origin in linguistic studies. This model, based on dialog acts, is clearly different from the task model. Such a distinction allows for a greater genericity. A hierarchical representation of the different dialog acts allows a refined dialog control and anticipation of exchanges, whilst leaving more initiative to the user. Several examples illustrate the different functionalities of such a model in a representative application for information requests for train schedules, fares and ticket reservation at the SNCF (French national railway company).

It should be noted that the research described by the authors on understanding and dialog modeling constitutes an essential precondition to a generalized use of spoken language dialog technologies in systems designed for the general public. The presented approaches led to several prototypes using different languages (French and English) for different information retrieval tasks (train and plane schedules, ticket reservation), which underlines the advantages of the proposed choices. One of the prototypes developed within the framework of the European project ESPRIT No 9075 Mask (Multimodal Multi-media Automated Service Kiosk) has been tested by more than 200 selected people in a Parisian train station.

The care of the authors to situate their approaches in an historical context and to compare them with the most outstanding international achievements in the field, not only makes this work a reference for subsequent developments, but also constitutes a solid basis for a thorough teaching in applied linguistics, cognitive sciences, artificial intelligence and engineering science.

Françoise NÉEL

Head "Speech Communication" Group (1985-1994)
Head Multimodal Communication Platform Project of the Human-Machine
Communication Department, LIMSI-CNRS, Orsay, France (1994-2000)

This page intentionally left blank

Chapter 1

INTRODUCTION

The circulation and use of information in all areas of life highly depends on the availability of computer networks that each day enclose the planet in tighter meshes. Radio, television, telephone, increasingly denser transport systems, the extension of telematics, satellite technology, and the computer - playing a central role - break up today's borders between humans and bring them together in a new world of communication. This world allows users to access an increasing number of different databases including images and sounds, text content and multiple information available at different sites all over the world.

Today it is not necessary any more to move to different locations in order to read a book, a document or to obtain any information. Navigating in multimedia hypertexts allows users to circulate in a world where information is accessible to everybody. In near future, it will be sufficient to talk to a machine in natural language in order to obtain train or airplane schedule information, to book a seat or to buy a ticket.

If we analyze a dialog of a telephone caller requesting information from an operator, we realize that the operator has certain physical (auditory, articulatory, etc.) and cognitive (comprehension, reasoning, etc.) capabilities. He is therefore able to hear and to understand, to contextually interpret the utterances of the caller, to seek the requested information from a terminal, to manage the communication appropriately and, finally, to provide a relevant response to the caller.

A human-machine interface represents an implementation of models for these understanding and dialog processes. To date, it seems impossible to build machines enabling a dialog with anybody about any subject. State-of-the-art technology is limited to interfaces able to communicate with a person using spoken natural language in order to provide the requested information within a limited application domain.

The presented work lies in the area of human-machine communication and particularly covers systems for information requests. These systems offer human-machine interaction using spoken natural language.

In the reminder of this introduction, we define the fundamental concepts of language and dialog, both constituting the necessary knowledge for human-machine spoken language dialog.

The second chapter provides an overview on the semantic case grammar representation. Well adapted to the processing of spoken language utterances, it allows the system to cope with the dislocated and asyntactic character of speech. The application of the case grammar is illustrated by a detailed description of the air travel information system L'ATIS.

The third and main chapter of this book describes the dialog processing of the system. After presentation of the applied terminology and some general information on task modeling, we discuss an example model that is based on task and plan structures. The task structure contains rules for the interpretation of user utterances, for the generation of commands towards the application back-end (e.g., a database), as well as for the generation of natural language responses to the user. The plan structure enables the system to detect task-related incoherencies. This structure conveys to the dialog model some indications or requirements (request for precision, for example) that influence the dialog flow. After a presentation of certain concepts issuing from analytical philosophy and linguistics, we review implementations of different dialog models, by classifying them into four groups: structural, logic, plan and task-oriented. On the basis of various real and simulated dialog corpora studies, we present a structural dialog model. It is based on the language act theory having its origin in analytical philosophy, and relying on the mathematical linguistic theory of language.

1. Language

Object of a particular scientific discipline, language is primarily a practical issue that fills each moment of our life, including dreams, elocution or writing. Language holds a social function which becomes obvious when it is used: normal communication (conversation, information, etc.), oratory (political, theoretical, scientific, etc. speeches) and literature (spoken language folklore, written literature, prose, poetry, song, theatre, etc.).

Furthermore, the language influences large areas of the human activity. And if, in the normal communication process, we use the language almost automatically without paying particular attention to its rules, orators and writers are constantly confronted with this process, and handle it with an implicit knowledge of its laws, that science certainly has not yet totally detected.

Historically famous Greek and Latin orators dazzled and subjugated the crowds. It is well known that not only content and ideas influenced the audience, but the technique used by the orators to transmit these ideas using the language. Since the era of Ancient Greece, history has strongly relied on eloquence and rhetoric: Socrates, Platon, etc., are the precursors.

Language is a human property that allows to express and to communicate opinions and thoughts using a system of vocal or graphic signs. The language itself is a system of vocal signs used by a community of individuals to express themselves and to communicate. The language represents a social aspect of the individual; it seems to obey the social laws which shall be recognized by all the members of the community. Common to everybody, the language becomes a *spoken* language, the carrier of a unique message. The term *discourse* reflects the role of language in communication.

Language may be materialized by a succession of articulated sounds, a network of written marks, or even by a set of gestures. In fact, language may be apprehended like a system of communication signs between individuals or different communities. It therefore constitutes a more general discipline, called *semiotics.* Several meaning systems seem to be able to co-exist without necessarily having language as their basis. Therefore, gestures, visual signs, as well as images, photography, cinema, painting and music may be considered as languages since they transmit on the basis of a specific code a message between two individuals or communities.

Language seems to be a particularly complex system in which different problems interfere. Given their complexity and diversity, language studies draw upon philosophy, anthropology, psychology, psychoanalysis, sociology, and various linguistic disciplines.

2. Dialog and Computer

Dialog is ubiquitous in our society; at all times we use dialogs to communicate, ask for a service, negotiate, dispute, joke, or even to lie and to mislead.

Ever since humans have been motivated to create machines that imitate their movements, functions and acts. The development of this increasingly complex type of machines, called *automaton*, seems to be based on a magic, religious, scientific or entertaining motivation. Therefore, throughout the centuries various mechanical automata have emerged with those of Jacques Vaucanson being the most known[1]. In this constant evolution towards the imitation of the human by more sophisticated machines, a major event marked our time, the appearance of the computer along with a new discipline called Artificial Intelligence (AI) causing much interest in a constantly growing community. The famous logician Turing preceded the AI by raising the question *is a machine able to think?* It may be answered with the famous Turing test. And it is not by simple coincidence that Turing suggests to establish a dialog between a machine or a human on the one hand, and a person willing to learn about his interlocutor on the other hand. Maintaining verbal exchanges proves intellectual capabilities that are usually attributed to humans. Turing has been perfectly aware of this fact.

Before investigating the field of human-machine communication, it seems appropriate to define certain fundamental terms. Interaction is the mutual influence of two people. Conversation represents a particular type of interaction, i.e., vocal interaction. Any nonverbal interaction does not relate to conversation. For now we define the term *dialog* as an exchange of verbal statements, uttered between two humans or between the human and the machine.

For quite a long time, computer science has been limited to study the use of programming or database access languages enabling to communicate with computers. The emerging field of microelectronics and the development of the computer technology enabled novice and inexperienced users to directly access computers without the support of computer specialists. We have therefore been witnessing the evolution of data processing applications along with a significant change of the human-machine interaction paradigms. Compared to the so-called *traditional* communication modes, including the manipulation of icons and text menus on computer screens and keyboards, the sequences of questions and answers or commands in an automated call-center, a real finalized co-operative dialog seems to be an alternative. The importance of a human-machine spoken language dialog that is as close as possible to spoken natural language seems needful, in the same manner as it now seems crucial to make information systems accessible to everybody.

3. Human-Machine Spoken Language Dialog

The spoken language introduces certain specificities into the dialog. These will be analyzed in the following sections. Before presenting the different knowledge sources that are necessary to a spoken language dialog system, we will introduce the rules for a flexible and spoken natural language dialog, as well as the architecture of a human-machine spoken language dialog system.

3.1 Speech and Human-Machine Interaction

The efficiency of spoken language is surprising: a non-experienced person is able to enter approximately 20 words using the keyboard, to write 24 words, and to utter on average 150 words per minute. Undoubtfully, speech represents the most natural way of communicatation. It enable hands-free eyes-free interaction and, in addition, allows to engage a spoken dialog via the telephone. Speech also allows to establish conversation in certain operational situations where time is a crucial factor, such as in the air traffic control domain. In other words, speech is the most popular way of communication in our everyday life.

Experiments carried out by Chapanis (1979) have clearly shown the advantages of spoken communication in the accomplishment of a task: speed and reliability of the task execution (compared to the use of the keyboard input and screen output), a more natural, easy and spontaneous communication mode, the possibility to interfere with other communication modes, enabling a multimodal communication.

In addition to these advantages, the use of speech seems crucial in certain situations to compensate for other human communication channels, if theses are either saturated (e.g., a communication between the pilot and the aircraft), or inoperational in case of an handicap, e.g., for blind persons. All these advantages justify research in the area of automatic speech recognition, as well as its integration in human-machine spoken language dialog.

3.2 Specifics of Spoken Language Dialog

There exist considerable differences between written and spoken language: the writer is able to think about the formulation of his sentence. He may modify it until complete satisfaction. Similarly the reader may read a sentence again in case of incomprehension or doubt. In turn, speech production errors may be corrected, but they cannnot be eliminated. They need to be corrected in real time, which introduces hesitation, repetition and self-correction phenomena.

Human-machine spoken language dialog differs from written dialog primarily due to the limitations of current speech recognition systems and the intrinsic structure of the spoken language dialog. The limitation of speech recognition systems may be explained by the non-deterministic character of the recognition process including difficulties to account for short and degraded messages (e.g., hesitations, interjections, etc.). This limitation introduces a disturbing parameter into the understanding of messages, and thus into the dialog flow.

The intrinsic characteristics of spoken dialog include the spontaneousness of utterances sometimes yielding a significant amount of redundant information, repetitions, self-corrections, hesitations, contradictions, and even tendencies to stop the interlocutor. They also include the non-grammatical structure of human utterances which is not only related to the spontaneousness of the utterance but also to the spoken natural language itself. Finally, they include clarification and/or reformulation sub-dialogs that depend on the limitations of the speech recognizer or the quality of the speech synthesis.

After having reviewed the specificities of the human-machine spoken language dialog, we now develop the rules for a natural and flexible dialog.

3.3 Rules for a Smooth Spoken Language Dialog

Compared to dialogs between humans, the human-machine communication constitutes a completely different interaction mode. The dialog turns are well respected, and interruptions practically do not exist. Nevertheless, in order to obtain flexible dialogs it seems necessary to establish a certain number of rules. In the following, we successively examine speech recognition, the management of the communication channel, the linguistic constraints, flexibility issues, the problems due to speaker adaptation and the meta-reasoning.

Speech recognition capabilities. Due to their non-determinism, speech recognizers introduce certain errors which may disturb the understanding process and, consequently, the human-machine dialog. In order to obtain an acceptable interaction, it seems necessary to use a speaker-independent recognizer that accounts for the various spoken language dialog phenomena, i.e., the hesitations, interjections, etc.

Communication channel management. The human-machine spoken language dialog does not leave, by definition, any trace. Therefore, to be able to manage the dialog, it seems necessary to generate system messages. These include, for example, the standby messages asking the interlocutor for patience (e.g., *please standby*), the restart messages (e.g., *I listen to you, please go on*) and messages to maintain the dialog that are useful for the communication (e.g., *do not cross, wait*).

Understand, interpret and deal with linguistic phenomena. In order to achieve a flexible communication, utterances need to be processed depending on their context. The utterance *I need the schedules for trains leaving tomorrow to Lille* causes different system reactions depending on whether it is an information request or a reply to a previous question. This implies that semantic utterance analysis requires a contextual interpretation prior to the extraction of its intrinsic content. Furthermore, it seems necessary to account for the various linguistic aspects, including the language coverage at the vocabulary level and the authorized syntactic forms, as well as the processing of linguistic phenomena including ellipses, anaphors, synonymies and allotaxies. Anaphors are references to a word or to a concept quoted in the course of the dialog. They may be expressed by a pronoun, a demonstrative adjective, etc. (e.g., *draw a square, color it in red*). Ellipses are incomplete sentences, that require context information to make sense (e.g., *draw a square and a rhombus*). Synonymy is defined as an equivalence of different words in terms of their meaning, the allotaxy indicates the equivalence of different syntactic expressions.

Flexibility. The interlocutor freely expresses himselves. This results in two types of mechanisms for dialog control. The first one should allow formulations that do not correspond to the syntactic constraints of the language. The second mechanism aims at according an active role to the interlocutor in the dialog, whilst providing sufficient guidance. This implies that the machine needs to be able to identify the general dialog topic, on the basis of which it infers the goal and the eventual plan of the interlocutor.

Adaptation to the interlocutor. Depending on the application, the response of the machine needs to be more or less adapted to the interlocutor or user. This property seems crucial in intelligent computer-assisted educational systems (EIAO). It seems important for interfaces accessible to the general public (accounting for the age and socio-cultural level of the interlocutor) and still interesting to be considered in expert support systems, whereby distinction should be made between the adaptation at the level of user expertise, and the adaptation in terms of familiarity with the system.

However, it seems impossible to determine the exact degree a machine needs to adapt to the human and, to what extent the user should accept the constraints imposed by the machine. However, linguistic studies have shown that humans tend to modify their linguistic behavior, by using stereotypes and by limiting the size of his vocabulary, without necessarily being aware of the language level nor the vocabulary manageable by the machine. It has also been found, that in spoken language, humans tend to adapt their rate/rhythm and elocutionary speed to the machine. It should be pointed out that only a few spoken language dialog systems currently take user adaptation phenomena into account.

Meta-reasoning. Although this topic remains a research area, a natural dialog interaction implies that up to a certain degree the computer is aware of its own knowledge and capabilities or, in other words, has a certain representation of itself. This enables the system to be aware of its limitations and, consequently, to appropriately react to those user questions it is unable to answer. Although there exist differences between written dialog and spoken language, their cognitive levels are closely related. Therefore if we do not focus on the speech recognition and communication channel, the high-level rules can be considered as valid for both written and spoken language dialog.

3.4 Functions of the Spoken Language Dialog

After having reviewed some general rules for a flexible and natural human-machine spoken language dialog we now discuss the functions a dialog system needs to assume.

Communication channel management. It includes the generation of system utterances to manage the dialog.

Understanding and contextual interpretation. Both functions correspond to the integration of the syntaxico-semantic representation (the literal meaning) of the current user utterance into the discursive dialog representation, by resolving ambiguities, anaphors and ellipses.

Generation and synthesis. Both consist in generating and synthesizing system utterances comparable to those recognizable and understandable by the system.

Inference mechanisms. The introduction of reasoning allows to go beyond the stage of simple question and answer systems and therefore to deal with more complex dialogs.

Predictions. They enable the support of low level processes, such as speech recognition and understanding. Since predictions make use of high-level knowledge (semantics, pragmatics) they contribute to the improvement of the dialog quality.

Dialog management. On the basis of the recognized and semantically interpreted utterance, the corresponding actions need to be triggered (i.e., asking and answering questions, etc.). This control function of the dialog manager requires an exhaustive representation of the end-to-end system and consequently an access to the various knowledge sources that will be analyzed in the following section.

3.5 Knowledge for Human-Machine Spoken Language Dialogs

Modeling human-machine spoken language dialog requires multiple knowledge about the language, dialog, task, user and the system itself. In general, distinction is made between the static knowledge that does not vary throughout the dialog, and the constantly evolving dynamic knowledge.

The static knowledge includes the following models:

Language model. Due to the technology limitations, state-of-the-art systems generally make use of distinct language models both for recognition and understanding (due to semantic and pragmatic aspects). The first model is used to determine the correct sequence of words on the basis of the phoneme string, whereas the second model is used to analyze and to interpret the user utterances. It should be noted that the understanding process makes frequently use of language models. These models may also be a basis for the response generation so that the system has identical linguistic abilities for both analysis and generation. The language model for the understanding process includes lexical, syntactic and semantic components.

Task model. This model includes the application-related knowledge, such as the manipulated objects or concepts, their interrelations, the inference rules that enable generating new knowledge, and the description of the task execution. The task model enables the system to interpret an utterance (spoken, written or multimodal[2]) in the dialog context in order to accomplish a given task.

Dialog model. It provides a general description of the different application-dependent situations. It enables the system to contextually interpret the user utterances and to predict the dialog flow as well as authorized deviations from the main topic of conversation.

User model. This model administrates the knowledge about the user at all levels of the understanding process: phonological variants, stereotyped formulations, the user's point of view of the machine, his expertise in the application domain, etc.

System model. Still at a rather preliminary state of research, this model describes the knowledge of the system about its proper communication capabilities and its competence limitations (speech recognition, mouse, numerical glove, etc.). This model is particularly important for multimodal dialog systems.

The dynamic knowledge about the language includes the following sources:

Task context. It completes the task model in the case of an evolutionary application. The context includes task knowledge which is likely to change during the dialog. It enables the system to raise certain ambiguities and to interpret the user utterances.

Dialog history. The exchanges between the user and the system, as well as their structures, are stored in a history. This allows inconsistency and speech recognition error detection to resolve anaphors, to process ellipses and, finally, to predict the subsequent system messages. The dialog history is necessary for a *real* dialog, able to understand more than simple questions and answers.

User model. In addition to the static user model, a dynamic model that evolves throughout the dialog and depends on the user utterances, goals, plans, etc., may be established. This model allows to adapt the dialog to the user by adopting adequate strategies, by altering style and level of the generated system utterances to those of the user and by choosing possible explanations. User modeling becomes especially relevant in EIAO systems, and depends on the degree of the user knowledge. In this particular case, the model keeps track of how this knowledge evolves over time.

System model. This model may also be updated throughout the dialog. It is subject to change depending on the state of the connected peripheral devices (e.g., activity recognition) and according to the knowledge of the system about its own capabilities and limitations.

4. Spoken Language Dialog System

An overview of a spoken language dialog system is shown in Figure 1.1. It contains components for speech recognition, semantic analysis, dialog management, an interface to database access and a system response generation component.

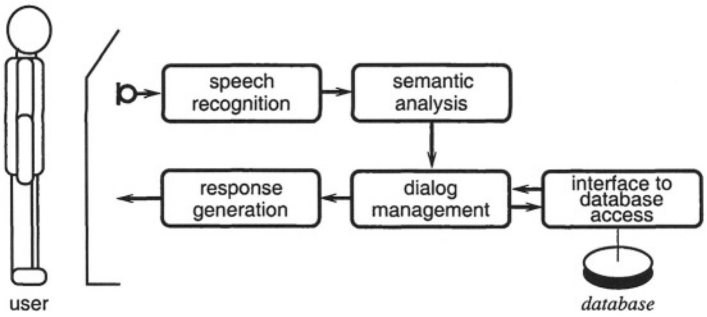


Figure 1.1. Spoken language dialog systems overview. It contains a speech recognizer, semantic analyzer, dialog manager, interface to database access and a system response generation component.

The input utterance is recognized by a speech recognizer (an introduction into the problem of speech recognition is given by Rabiner (1989) and Young (1992)). The output is then provided to the semantic analysis, which determines the meaning of the utterance and builds an appropriate semantic representation. Human-machine interaction, such as information retrieval, is a matter of interactive problem solving. The solution is often built up incrementally, with both the user and the computer playing active roles in the conversation. Contextual understanding consists of interpreting the user query in the context of the ongoing dialog, taking into account common sense and task domain knowledge. Semantic representations corresponding to the current utterance are completed using the dialog history in order to take into account all the information given by the user earlier in the dialog. If this information is insufficient for database access, ambiguous or if the database does not contain the information requested, the dialog manager may query the user for clarification and feedback. A database access interface uses the meaning representation to generate a database query and to access the application back-end, i.e., a database. A system response generator presents the interaction result in the form of text, speech, tables or graphics.

5. Projects and Research Applications

Several sites in the United States, Canada, Japan and Europe have been researching spoken language systems, including AT&T Bell Laboratories, Bolt

Beranek and Newman (BBN), Carnegie Mellon University (CMU), IBM Corporation, Massachusetts Institute of Technology (MIT) and the University of Rochester, United States, Centre de Recherche Informatique de Montréal (CRIM), Laval and McGill University, Canada, ATR, Canon, NTT and Toshiba in Japan, Centre National d'Études de Télécommunications (CNET) and Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur (LIMSI-CNRS) in France, Royal Institute of Technology (KTH), Sweden, Fluency Voice Technology (former VOCALIS) in Great Britain, DaimlerChrysler, FORWISS, Philips Research Laboratories and the University of Karlsruhe in Germany, and Centro Studi e Laboratori Telecomunicazioni (CSELT), Italy.

Several projects financed by the agency ARPA[3], in the United States, and by the European Union aim to support the deployment of technologies for speech recognition, spoken natural language understanding and dialog modeling. A significant number of applications relates to vocal servers and information terminals that provide train or airplane schedules for the general public.

ACCESS (European Language Engineering (LE) project) has been developing telephone call center technology coupled with recognition of written forms, received by post or facsimile. The call center handles standard insurance contract inquiries vehicle damage reports (Ehrlich et al., 1997).

ARISE (European LE project) aimed at building prototype automatic train schedule inquiry services to handle the bulk of routine telephone inquiries. One system has been developed for each of the Dutch and Italian railway operators and two for the French railway operator. The predecessor project, ***RAILTEL/MAS*** (European LE-MLAP project), defined the structure for the development of interactive voice services providing train timetable and scheduling in multiple languages (Dutch, English, French and Italian) over the telephone (Lamel et al., 1995).

ATIS (American ARPA project) was a designated common research task for data collection and evaluation support within the American ARPA Speech and Natural Language program. An ATIS system allows to the user to acquire information derived from the Official Airline Guide about fares and flight schedules available between a restricted set of cities within the United States and Canada (Price, 1990). L'ATIS is a French version of the American ARPA-ATIS task (Bennacef et al., 1994).

COMMUNICATOR (American DARPA project) aimed to support rapid and cost-effective development of multimodal spoken language dialog systems (Walker et al., 1999). The participating sites, mostly research laboratories, were required to use a common system architecture, i.e., the MIT-Galaxy Communicator Architecture (Seneff et al., 1998), by following a set of standards that promote interoperability and plug-and-play of components. The

COMMUNICATOR architecture supports development of sharable human-computer interface components for speech recognition and synthesis, dialog management, contextual interpretation, natural language understanding and generation. A shared research environment using the common task of travel planning, including, e.g., airline travel booking and car and hotel rental, common data, and a common evaluation framework, was created on a website, allowing developers to quickly assemble and test new architecture-compliant interfaces.

MASK (European ESPRIT project) has, as an application for information retrieval, much in common with ATIS. The project aimed at developing a multimodal, multimedia service kiosk in French to be located in train stations (Lamel et al., 1995).

SUNDIAL (European ESPRIT project) was concerned with handling information dialogs over the telephone. Four language prototypes have been developed for train timetable inquiries in German and Italian and flight inquiries in English and French (Peckham, 1993).

HOME (European Technology Initiative for Disabled and Elderly People (TIDE) project) is the development of an advanced remote control for a wide range of electronic home devices. The idea is to help elderly or disabled people with corresponding special needs as they feel overloaded with the large number of available functions and the variety of ways to get access to them. The input modalities used are vocal, tactile and gestural.

VODIS (European LE project) is developing a means of directing in-car devices using the human voice (Pouteau et al., 1997). The driver should be able to control an information system in German and French for working out the best routes for a journey.

Notes

[1] the flute player and the digesting duck.

[2] utterance generated by different communication channels, including speech recognition, mouse, gestures etc.

[3] ARPA (Advanced Research Projects Agency), now DARPA (for Defense), is one of the principal organizations of the Defense Department in the United States. It is invested in innovating search and the projects of development which have a significant potential for military and commercial applications.

Chapter 2

ROBUST SPOKEN NATURAL LANGUAGE UNDERSTANDING

1. Introduction

Nobody was able to foresee, 50 years ago, that the interaction between humans and machines would become increasingly sophisticated (high-level programming languages, multimedia graphic interfaces, etc.) and that such a huge number of people would use human-machine interfaces in the professional domain and also in their private lives. However, the main problem of dialog with computers relies in the difference between the *formal* languages, created to control the machines, and the *natural* language, used and understood by humans. This chapter is devoted to the way of how to fill this gap between the two types of languages. We will examine for this purpose a particular example of the current work on spoken natural language understanding.

Since for the most part, natural language research has its roots in symbolic system approaches, modeling of language understanding is often motivated by capturing cognitive processes, thus, integrating theories from linguistics and psychology. These cognitive models, however, are mainly established on the basis of written texts and often implemented using hand-crafted rules. Cognitive models presume the syntactic correctness of a sentence and in doing so, ignore spontaneous speech effects. The problem of ellipsis in spontaneous dialogs was analyzed by Morell (1988), but only few implementations deal with this issue in practice. Minor work has been dedicated to methods for recovery of interpretations in which parses are incomplete. (For example the utterance *how much time does it take in New York for limousine service* could be interpreted as the time either necessary to get a limousine at the airport or the transportation time between the airport and downtown New York.) Various analyses (Chapanis, 1979) considered spontaneous speech effects, including disfluencies, e.g., hesitations, repeated words and repairs or false starts, which are common in normal speech, as *afternoon flight from from Denver to San Franci- San Francisco.* Only a few research prototype systems, e.g., CMU-PHOENIX (Ward, 1994), take these effects into account. The ability to cope with spontaneous speech is crucial for the design of systems in real world applications.

The following sections, taken from (Minker et al., 1999), introduce the case grammar formalism, used by the system L'ATIS (Bennacef et al., 1994), as well as the train travel-based systems MASK (Gauvain et al., 1997) and ARISE (Lamel et al., 1998). Another example of a case grammar-based implementation is the CMU-PHOENIX parser (Ward, 1994).

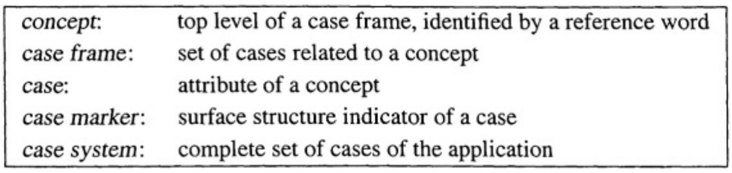
2. Case Grammar Formalism

In the domain of spoken language information retrieval, spontaneous effects in speech are very important. These include false starts, repetitions and ill-formed utterances. Thus, it would be improvident to base the semantic extraction exclusively on a syntactic analysis of the input utterance. Parsing failures due to ungrammatical syntactic constructs may be reduced if those phrases containing important semantic information could be extracted whilst ignoring the non-essential or redundant parts of the input utterance. Restarts and repeats frequently occur between the phrases. Poorly syntactical constructs often consist of well-formed phrases which are semantically correct.

One approach to extracting semantic information is based on *case frames.* A frame is a data structure, a type of knowledge representation in artificial intelligence (Minsky, 1975). It is a cluster of facts and objects that describe some typical object or situation, together with specific inference strategies for reasoning about the situation (Allen, 1988). A frame may be thought of as a network of nodes and relations. The top levels of a frame are fixed, and represent facts that are always true about the supposed situation. The lower levels have terminals or slots that need to be filled-in by specific instances of data. Each terminal can specify conditions its assignments must meet. The assignments themselves are usually smaller sub-frames. Collections of related frames are linked together into frame systems.

The original concept of a case frame as described by Fillmore (1968) is based on a set of universally applicable cases. They express the relationship between a verb and the related syntactic groups. Fillmore's cases correspond in fact to the Latin declensions: nominative, accusative and instrumental. Bruce (1975) extended the Fillmore theory to any concept-based system and defined an appropriate semantic grammar, whose formalism is given in Figure 2.1.

The case grammar uses in fact the stereotypical data structure of frames (Minsky, 1975). However, in order to fill in the frame slots, the notion of syntax (Fillmore, 1968) is added in the form of local marker-constraint relations. In the example query



concept:	top level of a case frame, identified by a reference word
case frame:	set of cases related to a concept
case:	attribute of a concept
case marker:	surface structure indicator of a case
case system:	complete set of cases of the application

Figure 2.1. **Semantic case grammar formalism (Minker et al., 1999).**

> *<you> <get> could you give me a ticket* **price** *on [uh] [throat_clear] a flight* ***first*** *class from* ***San Francisco*** *to* ***Dallas*** *please*

the case frame analysis identifies:

concept: airfare instantiated by the reference word *price*

cases: *San Francisco, Dallas, first*

markers: *from San Francisco* → from-city: *San Francisco*
to Dallas → to-city: *Dallas*
first class → flight-class: *first*

The parsing process based on a semantic case grammar considers less than 50% of the example query to be semantically meaningful. The hesitations and false starts are ignored. The approach therefore appears well suited for natural language understanding components where the need for semantic guidance in parsing is especially relevant.

Case grammars relying on Fillmore's theory or Bruce's extension were applied by Hayes et al. (1986) in the context of speech understanding in an electronic mail application and by Matrouf and Néel (1991) in an air traffic control prototype. The robust parsing in the PHOENIX (Issar and Ward, 1993) system is an implementation of a case grammar which relies on Recursive Transition Networks (RTNs) as the means for expressing slot grammars. The French natural language understanding components of L'ATIS (Bennacef et al., 1994), MASK (Gauvain et al., 1997) and ARISE (Lamel et al., 1998) make use of a case grammar. The following section describes the L'ATIS system in more detail.

3. Case Grammar in the LIMSI-L'ATIS System

At LIMSI-CNRS, work on a French version of the ATIS task was initialized in 1993 in collaboration with the MIT-LCS Spoken Language Systems Group. The natural language understanding component of the MIT-ATIS system (Seneff, 1992) was ported to French, which enabled data to be collected with a WOZ (Wizard-of-Oz) setup (Bonneau-Maynard et al., 1993). Using the WOZ data, a natural language understanding component for the French system, L'ATIS, based on a case grammar (Bruce, 1975; Fillmore, 1968) was subsequently developed and integrated in a spoken language information retrieval system (Bennacef, 1995; Bennacef et al., 1994). The main system components following the speech recognizer are the natural language understanding component and components that handle database query and response generation.

The semantic analyzer carries out a case frame analysis to determine the meaning of the input utterance and builds an appropriate semantic frame representation. The dialog history is used by the dialog manager to complete the semantic frame. The database query generator uses the semantic frame to generate a database query, the SQL (System Query Language) command-sequence, which is passed to the database interface. It presents the result of the database query and an accompanying natural language response to the user.

The idea behind the understanding procedure used in (Bennacef, 1995; Bennacef et al., 1994) is not to verify the correct syntactic structure of the user utterance, but rather to extract its meaning using syntax as a local constraint. Therefore, in L'ATIS the predicate of the case frame is realized as a concept and not as a verb and the arguments are the constraints of that concept instead of adverbs and objects. In the utterance

> *donc je voudrais un vol de Denver à Atlanta qui parte euh dans 1'après-midi et je voudrais arriver à Atlanta euh autour aux alentours de cinq heures*
> (*well, I would like a flight from Denver to Atlanta that leaves uh in the afternoon and I would like to arrive in Atlanta about at approximately five hours*)

the predicate is the flight-concept and the constraints (cases) deal with departure city, arrival city and times.

Most of the work in developing the understanding component was defining the concepts that are meaningful for the task and the appropriate reference words. This undertaking, which is quite important (and difficult), is obviously task-dependent. However, in transferring to another task in the same domain, such as for train travel information and reservation as in use in MASK (Gauvain et al., 1997), many of the same concepts and reference words are conserved. In order to extract the possible case frame categories of the ATIS task and their cases, the French ATIS development corpus was manually analyzed. Five categories were identified (Bennacef, 1995; Bennacef et al., 1994) and are given in the form of concepts in Table 2.1. They are represented in the order in which they appear in the casual structure. The concepts related to inquiries for time and flight information are merged in a unique case frame flight, because the information returned in response to these types of user queries is the same. A set of 38 cases is used to represent the different information categories in all the case frames. These cases are classed according to different categories, e.g., itinerary, time, airfare. Case markers provide local syntactic constraints which are necessary to extract the meaning of the input utterance.

Table 2.1. Semantic concepts for the case grammar parser of L'ATIS (Minker et al., 1999).

Semantic concept	*Example reference words*	*Example utterance*
book	*choisis (choose),*	*Je* ***choisis*** *le vol numéro trois cent dix-sept*
	réserver (book)	*(I choose flight number three one seven)*
airfare	*coût (cost),*	*Je voudrais les* ***tarifs*** *des vols de Denver à Atlanta*
	tarifs (fare)	*(I want to know the fares for flights from Denver to Atlanta)*
aircraft	*genres (kinds),*	*Quel est le* ***type*** *d'avion pour la compagnie American*
	type (type)	*(What type of plane is used by American Airlines)*
stop	*arrêter (stop),*	*Dans quelle ville euh ce vol fait-il une* ***escale***
	escale (stopover)	*(In which city uh does this flight make a stopover)*
flight	*aller (go),*	*Je voudrais* ***aller*** *de Oakland à Denver*
	vol (flight)	*(I would like to go from Oakland to Denver)*

A declarative language containing a list of possible case frames and associated cases is used to describe the casual structure whose architecture is given in Figure 2.2. It contains the conceptual as well as common levels represented in intermediate and basic structures:

Conceptual level: The REFERENCE WORDS specify the words to instantiate the corresponding CASEFRAME during parsing. Sometimes the utterance may contain multiple concepts resulting in the generation of multiple frames. Utterance parsing is done by first selecting the corresponding case frame with triggering reference words. Then the slots of the frame are instantiated using the case markers whereby higher level structures make reference to lower level SUBFRAMEs. Pointers to these lower level SUBFRAMEs are labeled with the symbol @.

Intermediate level: It contains the marker-value relations expressing local syntactic constraints in the semantic case grammar. The SUBFRAME itinerary, for instance, contains the cases from-city and to-city. The words in brackets are the *pre-markers.* In *de* ***Denver*** *à* ***Atlanta*** (*from Denver to Atlanta*), the preposition *de* designates *Denver to be the departure town and à* designates *Atlanta* to be the arrival town. In the SUBFRAME depart-hour-minute, heures (*hours*) is used as a *post-marker.* Pre-markers which are not necessarily located adjacent to the case may provide information useful in determining the context of the case.

Conceptual level	CASEFRAME airfare {REFERENCE WORDS: prix (*price*), tarif (*fare*), ... itinerary: @itinerary. times: @times. ...}	
Intermediate level	SUBFRAME itinerary {from-city: [quitte (*leave*), de (*from*)] @city. to-city: [à (*to*), pour (*for*), vers (*towards*)] @city. ...}	*Common levels*
	SUBFRAME times {rel-departure: [part& (*leaves*), partir& (*leave*), arriver! (*arrive*)] @comparative. departure: @depart-hour-minute. ...}	
	SUBFRAME depart-hour-minute {depart-hour: [part&, partir&, arriver!] @hour [heures (*hours*)]. depart-hour: [part&, partir&, arriver!] @noon-midnight. depart-minute: [part&, partir&, arriver!, heures, midi (*noon*), minuit (*midnight*)] @minute. ...}	
Basic level	SUBFRAME city {city: Denver, Boston, Atlanta, ...}	
	SUBFRAME comparative {comparative: avant (*before*), après (*after*)}	
	SUBFRAME hour {hour: 1,2,3, ...}	
	SUBFRAME noon-midnight {noon-midnight: midi, minuit}	
	SUBFRAME minute {...}	

Figure 2.2. Casual structure (Bennacef, 1995; Bennacef et al., 1994) for the natural language understanding component of L'ATIS. It contains the conceptual level as well as intermediate and basic structures in declarative form. The symbol @ refers to lower level frames, & designates non-adjacent markers. The exclamation mark ! designates the word as a non-marker.

In *qui part vers* ***vingt-deux*** *heures trente* (*that leaves by twenty two hours thirty*), the value *vingt-deux* corresponds to the case depart-hour, because it is preceded, although not directly, by the marker *part*. In the case depart-hour, *part* is therefore used as a *non-adjacent marker* (part&). In *partir euh cet après midi et euh je dois arriver le plus près possible de* ***dix-sept*** *heures* (*leave in the afternoon and I need to arrive closest possible to seventeen o'clock*), the value *dix-sept* is an arrival time in the SUBFRAME arrive-hour-minute. However, with the non-adjacent marker *partir* in the SUBFRAME depart-hour, *dix-sept* could also be identified as a departure time. In order to avoid this parsing error, the *cumulative non-marker* (partir!) was introduced in arrive-hour-minute (respectively arrive! in the SUBFRAME depart-hour-minute). The flag ! prohibits the accompanying marker to precede the corresponding value. Table 2.2 illustrates the semantic marker types which are used in L'ATIS.

Table 2.2. Case marker types used in L'ATIS (Minker et al., 1999). The symbols & and ! designate the marker type in the declarative structure (Figure 2.2).

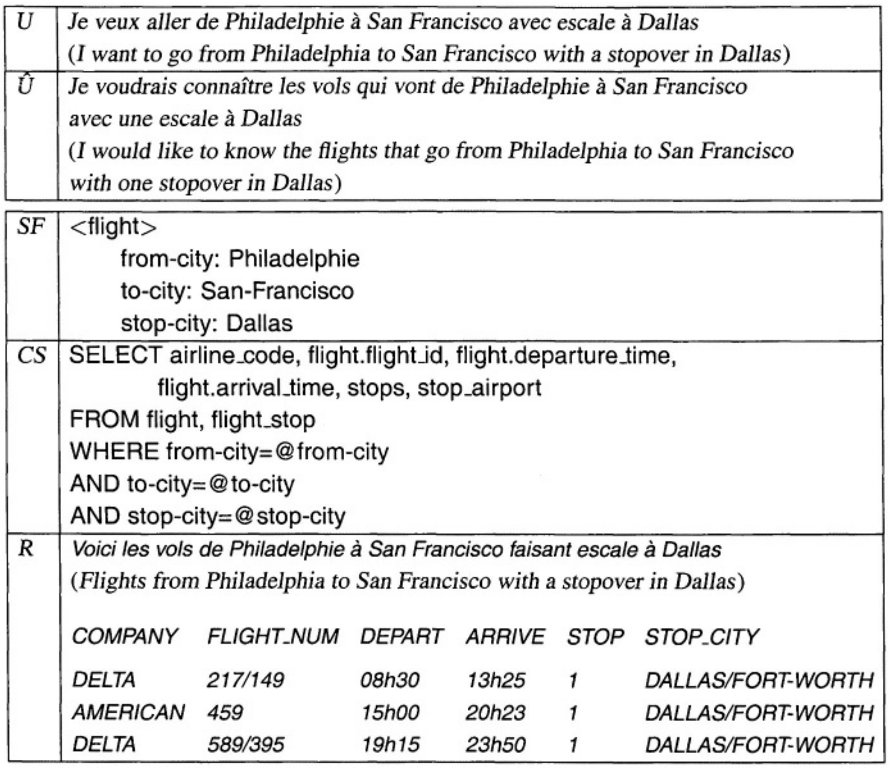
Marker type	*Example markers*	*Example utterance*
pre-marker	de (*from*), à (*to*)	*je veux aller de* ***Philadelphie*** *à* ***Dallas*** (*I want to travel from Philadelphia to Dallas*)
post-marker	*heures* (*hours*)	*celui qui part a* ***huit*** *heures le matin* (*the one leaving at eight o'clock in the morning*)
non-adjacent marker	*arriver&* (*arrive*)	*je voudrais arriver à Atlanta dans* ***l'après midi*** (*I would like to arrive in Atlanta in the afternoon*
non-marker	*partir!*	*je voudrais parler partir de Dallas et arriver euh en* ***fin de soirée*** *euh à San Francisco* (*I would like to leave from Dallas and arrive at the end of the evening in San Francisco*)

Basic level: It contains a list of authorized arguments or slot-fillers, like the SUBFRAMEs city and hour. They mainly correspond to values in the relational database.

The parser is recursively applied on the SUBFRAMEs until there are no suitable words left to fill in the slots. The case markers are successively removed from the utterance after the case instantiation. Once completed, the semantic frame(s) represent(s) the meaning of the input utterance.

Figure 2.3 illustrates the structures used at different analysis stages for parsing, SQL command-sequence and response generation.

For the example utterance *U*, the reference word *aller* (*go*) causes the parser to select the case frame flight. The complete semantic frame representation



U	*Je veux aller de Philadelphie à San Francisco avec escale à Dallas* *(I want to go from Philadelphia to San Francisco with a stopover in Dallas)*
Û	*Je voudrais connaître les vols qui vont de Philadelphie à San Francisco avec une escale à Dallas* *(I would like to know the flights that go from Philadelphia to San Francisco with one stopover in Dallas)*

SF

```
<flight>
     from-city: Philadelphie
     to-city: San-Francisco
     stop-city: Dallas
```

CS

```
SELECT airline_code, flight.flight_id, flight.departure_time,
          flight.arrival_time, stops, stop_airport
FROM flight, flight_stop
WHERE from-city=@from-city
AND to-city=@to-city
AND stop-city=@stop-city
```

R

Voici les vols de Philadelphie à San Francisco faisant escale à Dallas
(Flights from Philadelphia to San Francisco with a stopover in Dallas)

COMPANY	*FLIGHT_NUM*	*DEPART*	*ARRIVE*	*STOP*	*STOP_CITY*
DELTA	*217/149*	*08h30*	*13h25*	*1*	*DALLAS/FORT-WORTH*
AMERICAN	*459*	*15h00*	*20h23*	*1*	*DALLAS/FORT-WORTH*
DELTA	*589/395*	*19h15*	*23h50*	*1*	*DALLAS/FORT-WORTH*

Figure 2.3. Example L'ATIS utterances (*U*,*Û*), corresponding semantic frame (*SF*), SQL command sequence (*CS*) and formatted system response (*R*) (taken from (Néel et al., 1996)).

SF is constructed by instantiating the slots from-city, to-city and stop-city with the corresponding words *Philadelphie, San Francisco* and *Dallas* respectively. The analysis is driven by the order of the cases appearing in the case frame flight. Another utterance *Û* results in the same representation as that of utterance *U* since the reference words and markers trigger an identical frame and identical cases. The database query generator constructs the database query *CS* in the form of an SQL command sequence for database access.

The SQL command sequence is built from the semantic frame using specific rules, where each rule constitutes a part of the SQL command. In the example, the SQL command SELECT airline_code, flight_id, departure_time, arrival_time, stops, stop_airport FROM flight is produced on the basis of the semantic frame flight of the utterances *U* or *Û*. If the slots @from-city, @to-city and @stop-city contain values, WHERE from-city= @from-city AND to-city= @to-city AND stop-city= @stop-city is concatenated to the SQL command. It takes the appropriate values for from-city, to-city and stop-city from the semantic frame (*Philadelphie, San Francisco* and *Dallas* respectively). The rules for parsing an SQL command generation are defined in declarative form in order to allow for easy modification and flexibility. Once generated, the SQL command sequence is used to access the database and retrieve information. This information is then reformatted for presentation to the user along with an accompanying natural language response *R* which may optionally be synthesized.

4. Conclusion

The understanding component of L'ATIS makes use of a case frame approach to extract the meaning of a spoken input utterance. This semantic grammar formalism is considered to be more suitable for applications dealing with spontaneous human-machine interaction than formalisms based on a purely syntactic analysis such as formal context-free grammars. During parsing, syntax is only used to a minor degree enabling the method to be robust facing natural language effects. The *a priori* semantic knowledge gained from the manual analysis of a large number of utterances is expressed as a system of rules in declarative form.

This page intentionally left blank

Chapter 3

SPOKEN LANGUAGE DIALOG MODELING

1. Introduction

This chapter is devoted to human-machine spoken language dialog modeling (cf. the stages of dialog management, interface to database access and system response generation shown in Figure 1.1). After presentation of the terminology and an introduction to task modeling, we describe an example model that is based on task and plan structures.

The task structure contains utterance interpretation rules, rules for generating commands to the application back-end (e.g., a database) as well as the generation rules for natural language responses to the user.

The plan structure allows to detect task-related inconsistencies. It transmits indications or requirements to the dialog model (e.g., requests for precision) that influence the dialog flow.

After a presentation of the concepts introduced by the analytical philosophy and linguistics, we review different dialog models and classify them into four groups: structural, logic, plan-oriented and task-oriented. After a study of simulated and real dialog corpora we present a structural dialog model based on the theory of language acts. It has been introduced by the analytical philosophy and is also based on the theory of languages resulting from mathematical linguistics.

2. Task Modeling

In the following, we describe the task modeling in the domain of information requests.

2.1 Task Concept

State-of-the-art spoken language dialog systems cover a well-defined application and perform several tasks within that application. The system may be viewed as an interface between the user and the application. It accounts for user actions and translates them into specific tasks. For example, in a dialog system for robot control, the user may command the robot using spoken dialog to perform tasks like moving or raising an object. Similarly, in a graphic design program, the user draws more or less complex figures assisted by a spoken language dialog interface. Another example are spoken language dialog systems for information retrieval that enable a database research on the basis of user requests.

A task may therefore be defined as a succession of actions that potentially fulfill one or more user aims. An activity is defined as a succession of actions performed by the user within a specific task (Caelen, 1994). By a task model, we subsume the modeling of various types of knowledge for a given application. This knowledge may contain the manipulated objects or concepts, their interrelations, the inference rules that enable the system to generate new knowledge, the description of the task execution which is similar to the description of the scenarios[1] (Ferrari, 1994; Ferrari, 1997) and, finally, the modeling of the objectives. This latter one enables the system to interpret a user utterance (spoken, written or multimodal) in the context of the dialog, in order to perform a specific action or task.

The application is generally simulated by an application back-end. The dialog system needs to understand the uttered user message and to encode the interpretation result, e.g., using SQL command sequences, for transmission to the back-end (e.g., a Database Management System - DBMS). The response is also transcoded to enable the dialog system to generate an appropriate feedback to the user.

In general, we may distinguish two types of application back-ends: objects that belong to *static back-ends* are time invariant except the user decides differently. In *dynamic back-ends,* the objects evolve user-independently over time.

Two examples may illustrate these issues. If, for example, the application back-end is a DBMS, where only the user is authorized to introduce modifications, we consider this as a static back-end. In an air traffic simulator the aircrafts move in the air sector without requiring any user intervention. Such a back-end is therefore dynamic.

Similarly, in the *static task model,* the knowledge is described without any possibility of evolution. In turn, in a *dynamic model* the knowledge may vary depending on the task, the user, or the course of the dialog.

In general, application-related information is shared between the application back-end and the task model. The former contains data that do not depend on the dialog system. The latter contains knowledge relative to the dialog flow. Certain information of the application back-end may be duplicated in the task model for efficiency reasons.

A typical example that illustrates the differences between the application back-end and the task model is a DBMS. It provides a data model for a specific application domain (e.g., air transportation information services). In this case, the task model contains not only the application-related knowledge that is not contained in the database. It also describes the way of how a particular task is executed (e.g., book a seat, buy a plane ticket, etc.). Should certain information in the database be frequently used by the dialog system, it should be duplicated in the task model in order to avoid frequent database access that may lead to considerable latencies.

2.1.1 Task Classification

We distinguish between *explicit* and *implicit* tasks. The first one requires a detailed description to allow a sufficient control, e.g., for process control applications. The second task type may be described by a complete set of constraints according more flexibility to the user, as for example required in Computer-Aided Design (CAD) tasks (Caelen, 1994).

In addition to the rather general task type definition, we are able to provide a more detailed classification based on the underlying aims of the task completion.

Learning. Knowledge acquisition by the user is subsumed under teaching or educational tasks (computer-assisted tutoring systems, air controller training, etc.). In turn, a system that learns from knowledge, performs a knowledge acquisition task.

Information. The user asks for information in a specific domain (e.g., information request on train or aircraft schedules).

Command. The aim of the user is to handle objects in a reference world (e.g., robot control, manipulation of a graphical design tool, etc.).

Assistance. In certain applications, the user needs to be assisted in decision processes, e.g., in the medical domain, where the doctor may get support by an assistant that helps in diagnosis and establishing a medication plan.

It should be noted that the transitions between the different types of tasks are not neat. For example, an information that supports the user in a decision process may be an information or assistance task.

2.1.2 Task Modeling Approaches

Approaches to task modeling have been investigated in different domains. In the following we present three classes of task models (Ermine, 1993).

The *functional* class of task models is centered around the functions and functionalities available to the user. The *structural* class focuses on the task description. Finally, a *mixed* approach combines structural and functional aspects.

Functional approach. In this approach, both task representation and management focus on the different functionalities of the system in which the tasks operate. A typical example is the modeling of actions a user is able to perform on the system by using a multimodal interface that refers to a set of input and output peripherals. It carries out a number of actions and controls their operations. The functional approach supports the description of functions that are available to the user. The functional task description is generally performed on the basis of formal grammars or Augmented Transition Networks (ATNs). Formal grammars are easily implementable models and provide a modular description tool.

Structural approach. In this approach, the formalisms are centered around the task itself and oriented towards problem solving. The aim is to describe the tasks by their objectives, the problem solving solutions, etc. To resolve a problem, it is broken down into several elementary actions. While the functional approach addresses the problem from the point of view of the user, the structural approach is centered around the actions of the machine.

Mixed approach. Certain task description formalisms combine both functional and structural approaches, namely the task representation to fulfill different functions and the representation of those actions that are adapted to the problem solution.

In an air traffic control application (Matrouf et al., 1990b), the application system is the air traffic simulator. Its role is to simulate the aircraft movements in an air sector. The system contains the dynamic parameter values of the aircraft including level, direction and speed. In this application the task model contains two types of knowledge. These are on the one hand some information about the aircrafts in the air sector as well as their parameters. The other knowledge type allows to model the task concepts in the form of frames containing semantic-pragmatic information, the constraints on the aircraft parameters, the rules for generating messages to the user or the simulator, as well as rules for the concept prediction.

2.2 Task Modeling in an Application for Information Requests

The task modeling presented in the following is based on a categorization of concepts performed by the semantic case grammar (cf. chapter 2). Before providing an in-depth description, we will present examples of the L'ATIS application to illustrate the required knowledge for an information retrieval system.

Consider a traveler who wants to obtain some information on flights going from *Denver* to *Boston* at a travel agency. He asks *which are the flights going from Denver to Boston on Monday morning?* The agent, using a terminal for flight schedules and airfares, provides to the system the departure and arrival city names, the departure day as well as the eventual time constraints *between 6 am and noon* for example, that he could infer from the time range *the morning.*

Assume that another traveler comes to the travel agency of the *Denver* airport requesting *I want a return flight for Boston.* It seems obvious that the agent will sell him a ticket for a flight from *Denver* to *Boston*, even though the departure city has not been explicitly. By default, *Denver* is selected to be the departure city.

If, for example, the traveler asks *which are the flights going from Boston to Atlanta with a stopover in Boston?* the agent detects the inconsistency in this utterance at the level of the layover city even before launching the system to search a flight connection.

To the request *I would like to book a seat the next week* the agent will reply *which city do you leave? at what time? in which class?* etc., in order to gather the necessary information for booking a seat.

We realize that the travel agent using the information system, has competences and task-related knowledge that should enable him to carry out the actions summarized in Figure 3.1.

1. Understand the user requests.
2. a. Interpret the knowledge on the basis of what has been said.
   b. Choose the default values depending on the context.
   c. Detect the inconsistencies.
   d. Ask the necessary questions given a certain order for the realization of the particular task.
   e. Formulate a request to the application back-end for information retrieval.
   f. Provide a clear and concise response to the user.

Figure 3.1. Knowledge of the agent providing information retrieved from a database.

The process of understanding user requests (point 1) is modeled in our example by the semantic case grammar (cf. chapter 2). The knowledge (point 2) is represented in the task model according to *task* and *plan structures.* The task structure contains a complete set of task-related rules enabling the system to interpret the user utterances, to choose default values, to generate commands towards the application back-end, e.g., the SQL command sequences in Figure 2.3, and to generate a feedback to the user. The plan structure describes the task execution in the form of a plan hierarchy.

2.3 Discussion

The task model presented in the context of the L'ATIS application is complementary to the understanding mechanism. It enables a contextual interpretation of user utterances and an eventual correction of semantic representations. The model should also calculate default values, transmit information to the dialog model in order to generate the appropriate user feedback according to a library of predefined plans. Furthermore the model should generate commands towards the application back-end and, finally, indicate to the dialog model the contents of system responses to be generated.

We distinguish between the dialog planning that enables the identification of user intentions, and the system action planning. The latter allows to obtain, confirm or correct certain task-execution parameters.

The task model, directly related to the application, should be separated from the dialog model that describes the dialog characteristics for a given class of applications. However, both models are not entirely independent. The task model communicates certain information to the dialog model. Should, for example, the system need to obtain or confirm certain task-execution parameters, the task model generates a dialog act of request for precision or confirmation which is managed by the dialog model.

It may be argued that the task model already includes a dialog model since it enables the system to carry out a dialog in order to generate a command to the application back-end and then a feedback to the user. In fact, the task model may be sufficient for question and answer systems yielding rather poor dialog capabilities. However, how to cope with a situation where the user does not directly answer the question of the system, but asks another question instead? How to detect that the user replies to a question he has been asked for by the system? How to dynamically modify the strategies of the system according to the dialog situation?

We attempt to provide answers to these issues by describing a dialog model that offers a formal framework, for the more advanced aspects of the human-machine spoken language dialog.

3. Human-Machine Spoken Language Dialog Modeling

The spoken human-machine spoken language dialog is challenging since it somehow integrates the machine like a real human partner into the communication process. Therefore, the machine needs to understand the dialog in which it participates in order to collaborate to the user task as best as possible. The machine should be able to interpret the relevant user utterances, i.e., the semantic representations the user wants to communicate.

Based on the fact that the language is not only used to represent the world, but also to incite actions, we place ourselves in the framework of an action theory. We first present the dialog act theory from a philosophical and linguistic point of view. We then review approaches to dialog modeling used in data processing. These can be classed into four groups: structural, logic, plan-oriented and task-directed models. We then analyze several real and simulated dialog corpora to derive a structural dialog model developed on the basis of the language act theory. This theory originates from the analytical philosophy, and the theory of the languages which itself results from mathematical linguistics. The application framework is that of information retrieval systems.

3.1 Language Act Theory

Before discussing dialog modeling issues, we present the language act theory.

Language Acts according to Austin. Austin, one of the predominant figures from analytical philosophy, criticized that philosophers, in general, do not consider all the available facts before seeking a solution to a problem (Austin, 1962). He showed that, if the suggested solutions are not satisfying, this may be due to a limited knowledge about the problem facts. Austin therefore investigated a more clear approach of describing and defining these facts, the philosophers could then base their work on.

The originality of Austin's method relies in the use of ordinary language to obtain the facts.

The ordinary language contains invaluable expressions: it incorporates all the human knowledge gathered throughout the centuries, as well as the various interdependencies established throughout generations. Austin has therefore studied the nature of language and everything that may be accomplished by speech in order to be able to apply his results to philosophical problems. He distinguishes between *constative* and *performative* utterances by examining their impact:

Constative utterance. It is an assertion that is mostly conceived as a correct or an incorrect description of the facts.

Performative utterance. It is evaluated in terms of success or failure and enables accomplishment of an action using spoken language.

If we analyze the utterances *the globe is round* and *the meeting is open* both yield the same semantic structure (argument and predicate), but they do not yield the same function. The first utterance represents a simple description (constative), whereas the second one incites an action (performative), i.e., that of opening the meeting.

Comparing the utterances *the meeting is open* and *the meeting is open in a great confusion* it seems that the second utterance is not performative due to the modifying phrase *in a great confusion.*

Due to its inconsistency, Austin replaced the performativity concept by the more general and more abstract concept of *language acts.* He considers any utterance to be primarily a language act:

Locutionary act. It corresponds to the generation of words that belong to a vocabulary and a grammar, and to which a meaning in the classical sense can be associated.

Illocutionary act. It defines how the words can be understood (the same words may be taken as an advice, a command or a threat).

Perlocutionary act. It provides room for the interlocutor's purposes (an evasive promise may be understood as a confirmation).

Austin focused on the illocutionary speech act, because he considered it to be the most essential one.

Searle (1969) described a structure of the illocutionary act and developed an illocutionary act classification based on twelve criteria. We retain only the most significant ones.

Structure of the illucutionary act. The distinction between locutionary and illocutionary acts may give rise to the presumption that, based on its internal structure, a language act should contain two components, namely its propositional content and illocutionary force.

Consider the utterances *Jean smokes a lot* and *does Jean smoke a lot?* We note that both utterances allow to define different illocutionary acts, an assertion for the first utterance and a question for the second one. However, if these acts are characterized by their illocutionary force, a single act may be defined: the speaker refers to the same person *Jean* and attributes the same property, namely *smoking a lot.* The illocutionary act is therefore represented by the function $F(p)$, where F represents the illocutionary force and p the propositional content. Both utterances may be formalized as follows:

$F(p)$ = *assertion (Jean smokes a lot)*
$F(p)$ = *question (Jean smokes a lot).*

Criteria for the classification of illucutionary acts. Among the twelve classification criteria defined by Searle (1969), the following seem the most significant ones:

Illocutionary objective. It specifies the type of obligation committed by the speaker or interlocutor by uttering the act in question. The illocutionary objective is related to the illocutionary force, but does not merge with it. Therefore, the illocutionary objective of requests is identical to that of injunctions: both attempt to make the listener accomplish an action. However, it seems obvious that the illocutionary force differs in both situations. According to Searle, the illocutionary force concept is composed of several elements; even though undoubtfully the most significant one, the illocutionary objective represents only one of these elements.

Adjusting the words and the real world. The illocutionary objective of an utterance may consist of rendering their words (more exactly their prepositional content) in accordance with the world. Alternatively, the illocutionary objective may render the world in conformity with the words. The assertions belong to the first category, the promises and the requests to the second one.

Psychological state of the interlocutor. An affirmation implies beliefs, an order desires, a promise intentions, etc.

Propositional content. It indicates the contents of the illocutionary act.

These four criteria allowed Searle to distinguish five main types of illocutionary acts, namely *assertive* (assertion, information), *directive* (command, request, question, permission, injunction), *promissive* (promise, offers), *expressive* (congratulation, excuse, thanks, complaint, greeting) and *declarative* (declaration, judgment).

In the following, we briefly introduce the logic of the language acts. It enables the construction of a formal semantics for spoken natural language.

Logic of language acts. There exist two main competing tendencies in contemporary language philosophy: the *logic* approach studies how the language may be related to the world and focuses on the truth conditions of affirmative utterances. In contrast, the *ordinary* language approach studies how the language is used in conversation and focuses on the different types of language acts generated by the speaker when generating utterances.

Throughout the last decades, Vanderveken (1988), foundationist of the logic of language acts, aimed at making converge these two philosophical tendencies. He has developed the illocutionary logic to establish a formal semantics of natural language capable of characterizing simultaneously the truth-conditions and the illocutionary aspects of the utterance meaning. Unlike recent philosophical doctrines attempting to reduce the utterance meaning simply to its pure sense, Vanderveken considers that both language meaning and usage are logically related in the language structure. Therefore it seems inappropriate to analyze the linguistic meaning of an utterance without studying the illocutionary acts.

Vanderveken starts from the fundamental assumption that each fully analyzed utterance needs to contain an illocutionary force marker. In French, like in most other languages, the mode of the main verb, the syntactic type of the utterance, the word order, as well as the intonation and the punctuation marks depending on whether the utterance is spoken or is written, are features of illocutionary force markers. For example, the imperative mode determines that imperative utterances are used to give commands.

3.2 Linguistic Studies

In the Nineties, the language act theory yield a significant influence in many disciplines, not only in philosophy, linguistics and semiotics, but also in logic, artificial intelligence, cognitive psychology, law, computer science, and engineering. We now demonstrate the use of language acts in the domain of linguistics.

Language acts and linguistics. Pragmatic linguistics, based on the analytical philosophy on language acts and on the conversational standards (Grice, 1975), has focussed on three research areas: first on studies of various types of language acts, qualified by Austin as illocutionary acts, and their condition of use; second, on studies of the various linguistic tools, available to the speaker to communicate the language act; finally, on studies of language act sequences in the dialog.

Pragmatic linguistics is defined as the analysis of contextual user utterance meaning. It is not aimed at describing the meaning of the sentence or its semantics, but the function of the language act generated by the utterance. If the utterance constitutes the maximum syntactic and semantic units, the language act may be viewed as the minimal pragmatic unit.

In contrast to semantics, defining the sense of an utterance in terms of its truth conditions, the pragmatic linguistics, by defining the meaning of a language act by its communicative function, provides some idea of the sense based on the enunciative function of the language.

The language act yields different properties:

- It aims at realizing an action, i.e., an activity that transforms the reality.

- The actions in question, i.e., those initiated by the language, are a *command*, *promise, request, question, threat, warning, advice, etc.* The appropriate interpretation of the language act depends on whether the intentional character of the utterance is correctly recognized. For example, to understand what the speaker would like to say by *there is a large bull behind you* the interlocutor needs to recognize the intention of the speaker: did he simply want to inform his interlocutor about the presence of the bull, or in turn, did he want to inform him about an imminent danger?

- The language act is a conventional act: it has to satisfy a number of conditions of use, subsumed under *contextual appropriateness conditions.* These determine how the language act is adapted to its context. The conditions yield an influence on the circumstances and on the intentions of the speakers implied in the realization of the language act. If the conditions are not satisfied, they may cause failures.

- The language act is both *contextual* and *cotextual.* Thus, the context allows to decide whether the language act realized in *I will come tomorrow* is a promise, an information or a threat, similarly to whether it is appropriate to interpret *there is a draught* literally or to consider it as an insinuation. The role of cotext also seems important for the language act characterization. The cotext determines conditions of cotextual appropriateness, i.e., a number of conditions that determine the degree of appropriateness of the act in the discourse or conversation. Thus, the response *it is Monday* to the question *what time is it?* seems cotextually inappropriate, since it relates to a discourse object that does not coincide with the one introduced by the question. The cotext also represent a significant information for the interpretation of the language act.

Consequently, the key notion of pragmatics seems the concept of conditions of *con-cotextual appropriateness.* This notion is somewhat different from the concept of truth conditions.

The Geneva Model. Following the language act theory developed above (cf. section 3.1), we have introduced the discipline of pragmatic linguistics that may be used as a basis for the discourse analysis. It aims at describing the hierarchical structure of conversation using a *static* model, called the Geneva model. It has been developed by Roulet (1981), and then used by Moeschler (1985, 1989, 1991).

The *static* model is based on a hierarchical and functional structure of conversation. It enables an analysis of the model using a system of hierarchical units that maintain functional interrelations.

To illustrate this aspect, we create an analogy with the syntactic analysis of utterances. *Jean loves Mary* may be analyzed by distinguishing between the phrase level *(S),* the syntax level, nominal group and verbal group *(NG, VG),* and the level of lexical categories, noun, verb, etc. (*N, V,* etc.). These levels reflect the hierarchical relations between the different units. The example utterance may also be analyzed from a functional point of view. It contains a predicate connecting two arguments that yield the function of the grammatical subject *Jean* and the grammatical object *Mary.*

The first component of the Geneva model is the hierarchical component, built by three major conversational components:

Exchange. It is the smallest dialog unit that forms the interaction.

Intervention. It is the largest monologal unit that forms the exchange. It contains one or several language acts.

Language act. It is the smallest monologal unit that constitutes the intervention.

The second component of the model, entirely functional, aims at allocating functions to the hierarchical components. The originality of the Geneva model relies on the fact that it distinguishes the illocutionary from the interactive function of the utterance. In this model, the components of type *exchange* are composed of components that maintain the illocutionary functions, whereas the components of type *intervention* are composed of components that maintain the interactive functions.

In addition to the static model, the conversation is apprehended by a *dynamic* model, subject to the following conversational constraints:

Interactional constraints. Of social nature, they enable a smooth interaction in terms of opening, closing and repair.

Structural constraints. These are the constraints imposed by the conversation and its structure during the dialog flow.

Linkage constraints. These are not imposed by the conversational structure, but by the conversational components during the semantic interpretation. The linkage constraints may be expressed by the following conditions: thematic, propositional content, illocutionary and argumentative orientation.

On the basis of the hierarchical and functional analysis, the dynamic approach prompts the different components from the point of view of their capacity to satisfy or to impose interactional, structural or sequential conditions.

The static approach is primarily centered around the analysis of the relations between components, whereas the dynamic analysis examines these relations in terms of closing and continuation of the interaction.

Even though the Geneva dialog model has been conceived for discourse analysis and not for a computer-based implementation, adaptations of this model to dialog systems exist (Bilange, 1992; Vilnat and Nicaud, 1992).

3.3 Dialog Modeling Approaches

After the description of the language act theory from a philosophical point of view, we have demonstrated how the language act, smallest pragmatic component, is integrated in a linguistic dialog model for spoken language analysis. In the following, we present the approaches to dialog modeling that have been developed since the end of the Eighties, from a rather linguistic, but still operational point of view.

The dialog model provides a general description of the different application-related situations: request for information, repetition, confirmation, etc. It also specifies the relations between these situations. Three dialog modeling approaches may be distinguished: the *structural models* have their origins in linguistics; the *plan-oriented models* are mainly based on artificial intelligence and employ the notions of plan, planification and plan recognition; finally, the *logic models* use a modal logic to represent the mental attitude of the interlocutor and the reasoning induced by these attitudes.

We also present research efforts towards the realization of task-oriented dialogs (Deutsch, 1974) combining dialog and task modeling.

3.3.1 Structural Models

The structural models present the human-machine spoken language dialog in a hierarchical structure of subsequent utterances. These models account for the more or less complex conversational structures, and allow to derive the basic components (language acts).

In the main research on structural modeling, the dialog models are presented in the form of a formal grammar and expressed by scenarios.

In the rewriting rules, the symbols between accodances {} may be followed by the * or + signs. A non-terminal followed by * occurs zero or an infinite number of times, whereas, if followed by +, it occurs at least once. The slash / in the right members of the rules is equivalent to an *exclusive or*.

LOQUI is a system enabling database access for call center employees (Ostler, 1989; Wachtel, 1986). It is based on a hierarchy of language acts that are divided into three major groups: *requests* (four language acts), *assertions* (seventeen acts) and *comments* (four acts). A language act is considered to be a procedural label that acts on a propositional content.

The dialog model in LOQUI may be formalized by the grammar represented in Figure 3.2.

The model uses the following dialog units:

- conversation (C),
- dialog (D),

C	$=$	$\{D\}^{+}$
D	$=$	$\{E\}^{+}$
E	$=$	$M\ \{SD\}^{*}\ M\ \{SD\}^{*}$
SD	$=$	$\{SE\}^{+}$
SE	$=$	$M\ M$

Figure 3.2. LOQUI dialog model using the following dialog units: conversation (C); dialog (D); exchange (E); sub-dialog (SD); sub-exchange (SE); movement ou intervention (M), the smallest unit.

- exchange (*E*),
- sub-dialog (*SD*),
- sub-exchange (*SE*),
- movement or intervention (M), the smallest unit.

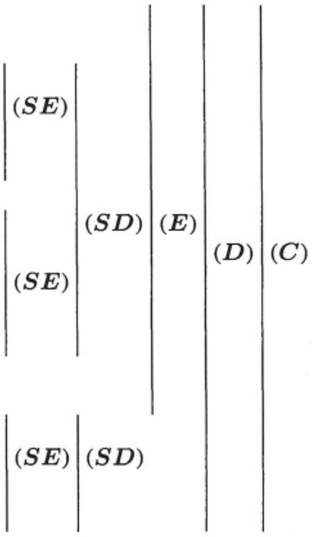
A conversation contains dialogs, each which consists of a series of exchanges. These exchanges may contain sub-dialogs, that in turn contain sub-exchanges. The dialog model is therefore represented as a hierarchical structure of interventions, exchanges, dialogs and conversation. The intervention is represented by an object yielding several attributes (the number of the intervention, agent, language act, topic, semantic content and dialog turn).

The following dialog example is based on the LOQUI model (the vertical lines group the different dialog units):

Example III. 1

(S : System, U : User)
U *: quand est-ce est le prochain IJACI ?* (***M***)
(when is the next IJACI?)
S *: je pense vous voulez dire IJCAI ?* (***M***)
(I think you mean IJCAI?)
U *: oui* (***M***)
(*yes*)

S *: vous voulez dire la prochaine conférence*
ou réunion ? (***M***)
(*do you mean the next conference or meeting?*)
U *: conférence* (***M***)
(conference)
S *: 12 août* (***M***)
(August 12th)
U *: 1987?* (***M***)
(1987?)
S *: oui* (***M***)
(*yes*)



Given the descriptive granularity of the language acts, problems may occur when transiting from a surface statement to a language act. The dialog grammar proposed in LOQUI imposes the relation between an utterance and a language act. This in turn considerably limits the possibilities of user expressions.

GÉORAL is a project supported by the GDR-PRC (Groupement de recherches coordonnées - Programme de recherche concerté (Coordinated Research Program) CHM (Communication homme-machine)) and directed by the computer science department of IRISA (Gavignet et al., 1992a; Gavignet et al., 1992b; Siroux et al., 1995). The aim of GÉORAL has been to develop a system for geographical and tourist database access using natural language. The retrieved database information is presented to the user in spoken and graphical form.

The dialog structure is fixed and modeled by a grammar whose terminals are dialog acts. The dialog flow performs the following steps: formulation of the request, negotiation of the request parameters, as well as negotiation of the system response and enumeration of the elements.

The GÉORAL dialog model may be formalized by the hierarchical grammar presented in Figure 3.3. It contains the following dialog units:

- conversation (C),
- dialog (D),

- exchange (E), contains an initiative movement (IM), an operational sequence of clarification sub-dialogs *(CSD),* an optional reactive movement *(RM),* and an optional sequence of evaluation sub-dialogs *(ESD),*
- movement (M), corresponds to a dialog act[2] *(DA),* an initiative movement that corresponds to a user request, whereas a reactive movement corresponds to a system response,
- sub-exchange *(SE),*
- sub-movement *(SM).*

C	$=$	$\{D\}^+$
D	$=$	$\{E\}^+$
E	$=$	$IM\ \{CSD\}^*\ RM\ \{ESD\}^*\ IM\ \{CSD\}^*\ \{ESD\}^*$
M	$=$	DA
CSD	$=$	$\{SE\}^+\ ESD =\ \{SE\}^+$
SE	$=$	$SM\ SM$
SM	$=$	DA

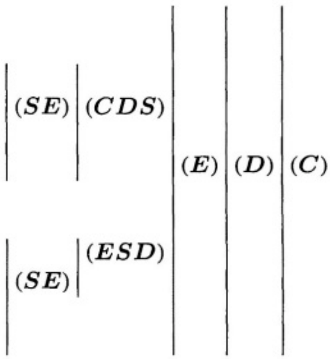
Figure 3.3. The GÉORAL dialog model. It contains the following dialog units: conversation (C); dialog (D); exchange (E), contains an initiative movement (IM), an operational sequence of clarification sub-dialogs (CSD), an optional reactive movement (RM), and an optional sequence of evaluation sub-dialogs (ESD); movement (M) corresponds to a dialog act (DA); sub-exchange (SE); sub-movement (SM).

The dialog acts may be divided into three categories: requests, responses, and communication management.

Example III.2

(U : User, S: System)

U : *je voudrais la liste des plages à Lannion* ***(IM)***
(I would like to have a list of beaches at Lannion)
S : *vous cherchez les plages à Lannion ?* ***(SM)***
(you are looking for the beaches at Lannion?)
U : *oui* ***(SM)***
(yes)
S : *il y a deux plages à Lannion* ***(RM)***
(There are two beaches at Lannion)
U : *montrez-moi la première* ***(SM)***
(Show me the first one)
S : *plage de Beg-Leger* ***(SM)***
(Beg-Leger beach)



Compared to LOQUI, GÉORAL uses a somewhat enriched dialog model. In fact, initiative and reactive movements as well as clarification and evaluation sub-dialogs have been introduced. The advantage of the GÉORAL model relies in its granularity. This is due to the distinction between initiative (request act) and reactive (response act) movements on the one hand, and the clarification and evaluation sub-dialogs on the other.

The Luzzati model distinguishes between three dialog levels, principal, secondary, incidental as well as their interrelations (Luzzati, 1995). The main level represents the thematic organization of the communication. The secondary level depends on the main one, since it is limited to the modification or addition of elements. The incidental level constitutes reformulation or precision requests following either a question or response.

The Luzzati model thus models the dialog on two axes: a horizontal *governing* axis, including the main question and response, and an *incidental* vertical axis, corresponding to the incidental question and response pairs. Formally, the dialog flow may be viewed as a horizontal and vertical graph construction.

STANDIA aimed at developing an intelligent telephone switchboard within the framework of GDR-PRC CHM. The system is able to process written and spoken language dialogs (Grau and Vilnat, 1997; Vilnat and Nicaud, 1992). STANDIA is based on the CARAMEL architecture, developed at LIMSI-CNRS (Sabah and Briffault, 1993). This architecture allows a dynamic planning of processes in a specific situation. As an intelligent telephone switchboard, STANDIA yields the following functionalities: re-direction of the caller to a given interlocutor, retrieval of information concerning the organization of the research institute or the current projects as well as the management of individual calendars and voice messages.

The main idea of STANDIA is to identify the user intentions in order to respond appropriately to his requests. Several modules enable the system to fulfill this task: the syntactic-semantic analyzer, the pragmatic interpreter (including thematic, intentional and interactional analyzers), the component for planning and generating language acts.

The thematic analyzer identifies any topic change. Based on the work of Luzzati (1995), the interactional analyzer ensures a co-operative behavior of the system. The intentional analyzer determines the language act that is related to the user utterance. Based on the Geneva model, the analyzer uses a hierarchical and functional dialog model.

Figure 3.4 presents the STANDIA dialog model with the following units:

- dialog (D),
- exchange (*E*),
- initiative (*I*); in *E I I I,* the first *I* yields the role of initiative, the second a reactive role and the third an evaluating role,

- simple intervention *(SI),*
- composed intervention (*CI*),
- language act *(LA).*

D	=	$E\ \{E\}^*$
E	=	$I\ I\ I \mid I\ I$
I	=	$SI \mid CI\ SI$
SI	=	$\{LA\}^*$
CI	=	$E\ I$
CI	=	$I\ E$

Figure 3.4. **The STANDIA dialog model with the following dialog units: dialog (D); exchange (E); initiative (I); in $E\ I\ I\ I$, the first I yields the role of initiative, the second a reactive role and the third an evaluating role; simple intervention (SI); composed intervention (CI); language act (LA).**

The originality of the STANDIA model relies in its capability to dissociate the language act label from the illocutionary function. Consequently, an interrogative act may yield the function of an illocutionary response. Depending on the dialog state and the expectations of the system, a system of rules determines, on the basis of the literal representation of the user utterance, the language act as well as its illocutionary function.

The STANDIA dialog model is sufficiently general to deal with different dialog types.

SUNDIAL has also been inspired by the linguistic Geneva model. The project concerns human-machine spoken language dialog modeling for database access in an airplane ticket reservation application (Bilange, 1991; Bilange, 1992). SUNDIAL proposes a structural and functional dialog model with four units: transactions, exchanges, interventions and dialog acts. The latter ones represent the elementary actions of the interaction. An intervention is built up by one or several dialog acts. It may yield an initiative, a reaction or an evaluation. The exchanges consist of interventions emitted by both dialog partners. They may contain the functional value of a reaction or of an evaluation. Finally, the transactions reflect the task resolution structure.

Figure 3.5 formalizes the dialog model that contains the following units:

- transaction *(T),*
- exchange *(E),*
- intervention *(It),* grouping initiative (*I*) , reaction *(R)* and evaluation *(EV),*
- dialog act *(DA).*

T	$=$	$\{E\}^+$
E	$=$	$\{It\ \{E\}^*\}^+ \mid \{E\}^+$
It	$=$	$\{DA\}^+$

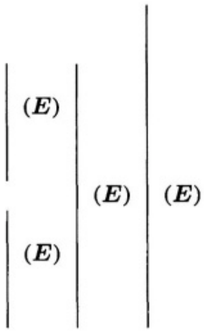
Figure 3.5. **The SUNDIAL dialog model containing the following dialog units: transaction (T); exchange (E); intervention (It), grouping initiative (I), reaction (R) and evaluation (EV); dialog act (DA).**

In the following examples, A and B denote two dialog partners:

Example III.3

(Simple case)
A : *je vous offre le vol de 10 heures 30. Ok ?* ***(I)***
(I propose you the 10h30 flight. Ok?)
B : *est-ce un vol TAT ?* ***(I)***
(is this a TAT flight?)
A : *oui* ***(R)***
(yes)

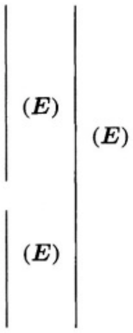
A : *à quelle heure arrive-t-il ?* ***(I)***
(at what time does it arrive?)
B : *à 11 heures 30* ***(R)***
(at 11h30)



Example III.4

(Evaluation case)
A : *quand voulez-vous quitter ?* ***(I)***
(when do you want to leave?)
B : *novembre 13* ***(R)***
(November 13th)
A : *novembre 13* ***(EV)***
(November 13th)

A : *quelle heure voulez-vous quitter ?* ***(I)***
(what time do you want to leave?)
B : *à 10h* ***(R)***
(at 10 o'clock)



The SUNDIAL dialog model, mainly based on linguistic studies, is descriptive. In fact, the grammar rules express structural constraints, and are limited to enumerate the possible subsequent dialog steps. For that purpose, the dialog model has been enriched by rules for control and conversation. These rules refine the dialog, exert a control and ensure its flexibility and naturalness.

VERBMOBIL is a German national project of interpretative dialogs for automatic spoken language translation. Like SUNDIAL, it uses four interrelated structural levels:

- dialog,
- politeness/negotiation formalities,
- dialog turn/exchange,
- dialog act.

These concepts do not directly relate to the ones used in SUNDIAL. A major difference relies in an additional formality level that accounts for *courtesy* and *negotiation.* It breaks down the dialog to several phases. The *intervention* level, defined in SUNDIAL is replaced in VERBMOBIL by a particularly rich and precise hierarchy of dialog acts. These are clearly centered around the application in question. The aim of the dialog is to negotiate an appointment (date and place) that is convenient to all the participants. A smooth categorization allows a refined control of the negotiation dialogs.

The VERBMOBIL dialog manager makes use of contextual knowledge. The recognition of the dialog act, for example *accept-date* or *reject-date,* seems appropriate for utterances like *ja, da habe ich keine Zeit (yes/well, I won't have time then).* In this example, the word *ja* should be translated by *well,* rather than by *yes.* The recognition and the prediction of dialog acts are performed by combining knowledge-based and stochastic methods. A plan recognizer groups all the dialog acts into phases, representing the complete dialog in an abbreviated form.

Information retrieval techniques allow the use of robust (surface) processing techniques. The main semantic content of the utterance, as well as information on the dialog act and date, are extracted from word lattice hypotheses. The translation is generated on the basis of models, in which the choice of the semantic frames corresponding to the utterance depends on the dialog act. Limited to a deep analysis, the utterance *ja, ich weil also würde mal sagen ehm vorschlagen, wir könnten uns am ehm siebten treffen so Mai (yes, I because then I would say hmm would like to propose, we could meet hmm May seven)* may be rejected by the system. No translation would be generated and the interlocutor would need to repeat the utterance. The surface processing identifies the dialog act as a *suggest-date,* then extracts information about the date on *May* 7 and generates the translation *what do you think about May 7?* The combination of both, surface and deep analyses, renders the system considerably more robust.

In addition to the generation of simple error messages such as *please speak louder,* the VERBMOBIL system is able to engage clarification dialogs with the user. The system recognizes errors in dates, e.g., *February 30* or *16 hours in the morning* and asks the interlocutor to provide an acceptable input.

As mentioned above, the hierarchy of the dialog acts in VERBMOBIL is application-driven and primarily concerns the negotiation dialogs. However, even though some acts are rather specific such as the date or place suggestion/rejection, the corresponding dialog act may easily be generalized to proposal/refusal or disagreement concepts. Furthermore, some acts are clearly task-independent such as the error correction, or the information feedback, which should enable this type of model to be applied in other domains.

Discussion. The linguistic research carried out at Geneva University, on which some of the described work is based, proposes a hierarchical and functional dialog model. It is aimed at creating a theoretical framework for dialog analysis. This assumes that the entire dialog is known prior to the analysis which does not apply to a human-machine spoken language dialog in which the conversation progressively evolves. However, the dynamic Geneva model accounts for the evolutionary aspect of the conversation, by imposing conversational constraints.

Several research groups have adapted the Geneva dialog model for operational purposes. Nevertheless, the hierarchy - exchange, intervention and language act - seems rather general for structuring the human-machine dialog. In fact, the grammar rules of the different models are not sufficient to direct and control the dialog. For example, the rule defining the exchange does not contain any information about the type of the intervention which is a part of this exchange. The existence of the initiative/reactive pair does not always seem sufficient: if two language acts overlap, it seems impossible to clearly determine to which initiative the reaction corresponds to.

3.3.2 Plan-oriented Models

A plan is defined as a sequence of operations transforming the world from an initial to a final state. In early planning systems, knowledge about planning processes has been represented in a procedural way. In current state-of-the-art systems, meta-plans, i.e., plans whose arguments are constituted in part of other plans, seem preferably used. This technique has been applied in particular by Wilensky (1981, 1983) for understanding tales. The dialog has been considered as an entity and the meta-plan principles been applied and adapted to dialog modeling. In general the plan types may be structured into several levels, as this can be observed in the Litman model.

The Litman model is based on three plan categories: the domain plans that model the application, the language acts that model the elementary communication actions and, finally, the discourse plans that model the relations between utterances and domain plans (Litman and Allen, 1987; Litman and Allen, 1990).

Earlier models are limited to rather local phenomena, by simply seeking to connect language acts and domain plans. The Litman model, however, accounts for global phenomena such as the management of clarification sub-dialogs by introducing the discourse plans.

The Litman model has also been used to deal with certain error situations concerning the quality of intentions the system associates to the user. Since these intentions are represented by a set of user plans, the system verifies the validity of the plans with respect to a reference stored in its knowledge base.

ATR developed a human-machine spoken language dialog system which predicts user utterances in different languages for a conference registration application (Yamoka and Iida, 1990). The dialogs are co-operative and task-oriented.

The understanding model is based on the plan recognition principle, which uses three domain-independent pragmatic knowledge sources and one domain-dependent knowledge source.

In the telephone dialog, the utterances may be categorized into *ordinary* expressions (communication act expressions and expressions yielding a propositional content) and *pre-determined* expressions.

The understanding model uses four plan types: *interaction, communication, dialog* (for building the domain-independent pragmatic knowledge), and, finally, *domain.* These plans are described on the basis of frames in a hierarchical way (interaction > communication > domain > dialog).

The ATR dialog component is able to associate to the corresponding communication act of the system-generated expression the act that is relative to the expected user utterance. Therefore, the act associated to *ask-value* is *inform-value.* In addition, the use of the propositional content refers to the concept that appears in the predicted utterance.

In the ATR system, distinction is made between domain-dependent and domain-independent knowledge. This enables to better analyze the specifics of human-machine spoken language dialogs.

Discussion. Plan-oriented modeling seems interesting, since it exhaustively describes the different dialog stages. These enable an appropriate interpretation of the user utterances and a detection of inconsistencies and errors in the dialog. However, the multi-plan approach that allows to distinguish between the dialog and domain planning seems most remarkable.

3.3.3 Logic Models

Logic models use a modal logic to represent mental attitudes of the interlocutor and the reasoning on these attitudes. Applications and systems based on these models are described in the following.

ARGOT is a system based on the language act theory including planning and user modeling, i.e., user intentions, abilities, knowledge, willingness, etc. (Allen and Perrault, 1980; Cohen and Perrault, 1979).

The language acts are considered as actions and the dialog as a planification process with the aim to determine the plan of the user, to help in its realization, etc. Generating a language act signifies performing an action to realize the plan. The dialog therefore consists of carrying out and recognizing plans.

Modeling agents. The modeling of beliefs, knowledge and goals of an agent is based on a modal logic system with the following operators (we mention only a few of them):

Want: *a wants x* signifies that the agent *a* wants *x*, where *x* is a well-formed formula in the model.

Can perform: *a can perform x* signifies that the agent *a* is able to perform the action *x*.

Believes**:** *a believes x* signifies that the agent *a* believes in the proposition *x*.

The following axioms apply:

1. If *believes p* ⇒ *q* and *a* believes *p,* then *a believes q.*
2. If *a believes p* and *a believes q,* then *a* believes *(p and q).*
3. *a knows p* ⇔ (*p* and *a* believes *p*).

Plans and actions. An action is defined by a frame yielding the following attributes: name of the action, parameters, preconditions, effects and body:

The action body for a language act is the utterance of the act. For example the language act *inform* whose three parameters are: *speaker*, *listener* and *proposition* is defined as:

Inform (l, a, p):

Precondition**:** *l* wants to inform (*l, a, p*) *&* *l* knows *p*.

Effect**:** *a knows p.*

An action should be considered either as physical (e.g., to take a train), or communicative (e.g., to inform).

Plans are consequences of actions. They connect an initial to a final state of the world. The planning process is built up of a unit of planning rules and a control strategy. An example for a rule is:

> *If an agent wants to reach a goal and the effect of an action is this goal, then the agent may wish to carry out this action.*

TENDUM is a system, in which the dialog is based on language acts which are functions acting on the context (Bunt, 1989). The context is limited to the mental state of both dialog partners. Due to the restrictions imposed by Austin's theory, Bunt uses the terms *dialog act* and *communicative function.*

The dialog analysis is performed by the following steps:

Syntactic-semantic analysis. The semantic content of an utterance is determined by the word meaning *(formal level)* and by the way in which the words are combined in the dialog context *(reference level).*

Pragmatic analysis. It does not determine the semantic content of the utterance, but its *communicative* function in the dialog. The combination of the semantic content and the communicative function is called *dialog act.* The communicative function is performed by identifying the surface form of the act and by interpreting this form by its role for the communication in the dialog.

Communicative functions and user modeling. The main communicative functions used in TENDUM are *questions* and *answers* (Bunt, 1994). Several question/answer types have been identified in order to determine the relation between the semantic content and the user knowledge and objectives. These relations have been expressed by epistemic operators. The communicative functions are hierarchically organized and therefore play an important role in the system since they build and maintain the knowledge model and objectives of the user. The dialog flow is therefore the result of a movement through the act hierarchy by adding preconditions to the dialog acts starting from the root down to the leaves. The preconditions are expressed by logic formulas derived from the epistemic logic.

Information evaluation and database access. This evaluation consists of controlling the consistency of the information related to the domain of discourse by accessing the database.

Dialog generation. TENDUM is based on the following idea: given a user goal or objective, generate an action that aims at satisfying this goal.

The system therefore generates a dialog act or accesses a planning component. It establishes the necessary plans to satisfy the user goal.

AGS-ARTIMIS (Audiotel guide des services / Agent rationnel à la base d'une théorie de l'interaction mise en oeuvre par un moteur d'inférence syntaxique) is a system in line with the tradition of logic. The research carried out by CNET (Sadek and Mori, 1997) is based on the assumption that problems in natural human-machine communication are not only related to algorithms. They require the restitution of such complex aptitudes like the perception and the production of language, understanding or reasoning.

The work at CNET is based on two concepts, namely the mental attitude (Searle, 1983) and the actions (Austin, 1962). The originality of the *AGS-ARTIMIS* approach, based on a logic model of mental attitudes of the action, relies in the construction of a solid methodological and theoretical framework, that defines in detail the basic concepts to manipulate. The research of the deep nature of the intention concept has led to a more general problem, that of modeling the rational balance[3]. It is maintained between the different mental attitudes of an agent and, on the other hand, his mental attitudes, plans and actions.

Two important ideas influence the CNET approach: the establishment and continuation of a dialog can be entirely justified by the principles of rational behavior. Furthermore, the same logic theory may account for various dynamic aspects underlying the rational behavior, whereby the intention plays acentral role in the control of this dynamics.

Discussion. The advantage of logic-oriented dialog modeling relies in the solid methodological scope, which deals with the fundamental aspects of interaction such as intentionality and rational balance. However, two limitations seem essential. First, such an approach cannot be deployed if the underlying concepts do not use a clearly specified and formalized model. The approach also requires a significant number of axioms and inferences, which prevent it from exerting its potential on those aspects to which other approaches still provide acceptable results.

An important issue in human-machine spoken language dialog models is related to their practical use: do we wish to use these models to develop effective dialog systems or do we aim at cognitively modeling the interaction in general. Practical tasks often do not require the level of detail of these cognitive approaches.

3.3.4 Task-oriented Models

Task-oriented models are closely related to the application. The knowledge about the dialog is combined with the task knowledge. In the following we describe example applications making use of this type of model.

MINDS is a spoken language dialog system for accessing a stock management database, developed at CMU (Young et al., 1989; Young and Ward, 1988). The task associated to each dialog scenario concerns damaged ships. The system provides information about the damage characteristics and determines if the ship is able to continue its mission or has to be replaced.

MINDS uses a set of knowledge to establish predictions used to narrow the search space for the speech recognition process. Three knowledge sources enable generating the predictions: domain knowledge, a tree of *and-or* goals representing the hierarchy of all the goals the user is supposed to express throughout the dialog and a user model.

In the MINDS system, no distinction is made between the dialog and the task model. Both are merged in the structure of goal trees. This structure seems well-suited for the development of plans. The predictions by layer seems rather effective, nevertheless burdensome due to the fact that each time the predictions fail, the analyzer needs to restart the analysis using a less constraining grammar. Furthermore, this method requires the specification of large data amounts (organization of the knowledge base, partitioning of grammar, linkage with the concepts, etc.).

VODIS (Voice Operated Database Inquiry System) developed at Cambridge University is a system for database access via the telephone (Proctor and Young, 1989; Young and Proctor, 1989). The recognition component is related to a frame-based dialog control.

This approach enables a strongly task-related dialog that depends on the speech recognition performances. The dialog strategy may be changed as a function of the speech recognition quality. Furthermore, a study of the production mechanisms allows the generation of identical contents by using different formalisms depending on the dialog context.

The aim of VODIS II has been to modify the syntactic constraints and the dialog type as a function of the speech recognition quality. Syntax (out of context) is defined in the form of uncompiled rules. Depending on the context, the system activates or deactivates the rules. The recognition process including syntactic constraints generates a lattice which is analyzed. The syntactic constraints are obtained using a semantic context-free grammar.

SYDOR developed by LIMSI-CNRS is a spoken language dialog system that is driven by the task between the user and an application back-end (Matrouf et al., 1990b). Different knowledge sources, including the vocabulary, the syntax, the task model, the user model and the dialog history, are used during the dialog. This knowledge is represented in a unified way by hierarchical frames.

SYDOR performs a two-stage utterance analysis. The categorization consists of determining the conceptual utterance category. The instantiation yields the role of collecting semantic information in the utterance. This information is then used to fill in a frame that corresponds to the given category. In order to account for the limitations imposed by the speech recognition component, but also for the particularities in the human interaction, detection and internal error correction mechanisms have been introduced. These make contribute all types of knowledge, including acoustic and semantic-pragmatic levels. In case of failure, a dialog protocol is established for error recovery.

The originality of SYDOR relies in the use of pragmatic knowledge to predict on the user utterances. This enables a dynamic limitation of the search space of the speech recognizer and an optimization of the user-system exchanges (Matrouf et al., 1990a; Matrouf and Néel, 1991). Compared to MINDS the use of the predictions is easier, since the probabilities are attributed to characteristic concepts and words. Consequently only one recognition step is required.

SYDOR, integrated in the industrial prototype PAROLE (Marque et al., 1993), has been evaluated within the framework of an air traffic control application. It manages a dialog between an air controller and an air traffic simulation system.

The system has been designed for training air controllers. A frame-based dialog model seems therefore appropriate for this type of relatively simple dialogs. However, the system is able to manage multiple dialogs with various simulated pilots in parallel.

DIAL of the Centre de recherche en informatique de Nancy (CRIN) is a dialog system with a strongly integrated architecture, where the knowledge of different levels contribute to the understanding of an utterance (Roussanaly and Pierrel, 1992). The general architecture is based on independent components: acoustic-phonetic decoding, prosodic processing, lexical processing, syntactic-semantic analyzers and dialog management.

From an operational point of view, the dialog processing is a communication mainly between the syntactic-semantic analyzers and the lexical processing. It is composed of three modules, the interpreter, the reasoner and the dialog component. The interpreter combines the information from other modules to generate a contextual interpretation. The reasoner proposes a set of solutions that are likely to satisfy the user request. The final decision of which response to generate to the user relies on the dialog module.

In DIAL, the dialog module is important, because it integrates different types of knowledge, including those related to the interpretation, to the task in the form of facts and rules, and those related to the dialog in the form of frames. The task-related knowledge representation seems quite interesting, since it represents the expertise required for problem solving.

Discussion. The task-oriented approach to dialog modeling is neither based on linguistic theories nor on the fundamental aspects of interaction. Rather pragmatically, it is oriented towards the realization of dialog systems.

3.4 Dialog Modeling in an Application for Information Request

In spoken language dialog systems, the dialog model is frequently merged with the task model. This is done either for effectiveness reasons, or because of the task characteristics which make the dialog not really discernible from the task. Nevertheless, at the cognitive level, humans yield entirely domain-independent dialog capabilities. A distinction between dialog and task model seems therefore appropriate.

3.4.1 Why a Dialog Model?

One of the most significant roles of the dialog model in a spoken language dialog system seems to appropriately represent a contextual interpretation of user utterances. This allows the system to generate the most adequate system response without limiting the dialog to a succession of questions and answers. This role should also enable the system to anticipate/predict, to raise ambiguities, to correct errors, to explain system decisions and to trigger the corresponding actions throughout the dialog in order to suitably manage other processing modules.

With an *a priori* knowledge about the dialog flow, the system is able to recognize more easily, for example, a precision or reformulation request, and to know at which interaction stage it may authorize such an intervention. For example, the utterance *I want to leave to Dallas* may be interpreted, depending on the dialog context, as a request for information, a confirmation, a response, etc. Another reason that justifies the use of a dialog model is the need to formalize the human-machine dialog. This may be performed by a distinction between the task and dialog model.

3.4.2 Why a Structural Model?

In the following we describe an example of a dialog model that is based on a well-established theory. Since there has always been research for invariant and atomic entities in terms of morphemes in morphology, lexemes in syntax and sememes in semantics, the language act may be viewed as the basic dialog unit. Inspired by the linguistic work on hierarchical dialog modeling (exchange, intervention and language act), and based on the grammar theory that defines a formal language system using rewriting rules, the dialog is modeled by a grammar whose non-terminals correspond to sub-dialogs and terminals to language acts.

Corpus analysis. We observe two opposite tendencies in social sciences. For some researchers, the knowledge acquisition depends on the observation of the human behavior. For others, these observations make sense only if they reveal the subjacent laws of human behavior and allow us to derive the corresponding models.

The dialog modeling approach presented in the following is situated at the crossroads of both tendencies: it starts from theoretical models, i.e., the linguistic dialog models, to analyze dialog corpora for different applications that have been recorded under different conditions and are based on different protocols. Such a corpora study allows to analyze the different phenomena that may occur in the dialog. However, even though these analyses are rarely exhaustive, it seems necessary to permanently ensure the generality of the gathered information.

With this intention in mind, several corpora have been analyzed by extrapolating non-observed situations.

***SNCF* Corpus.** Within the framework of a GRECO (Grenoble campus ouvert) research experience, a telephone corpus of user utterances has been recorded in the SNCF information center at the Saint Lazare train station in Paris (Luzzati, 1995; Morell, 1988). The recordings have been made with motivated users calling at the center. The aim has been to analyze the linguistic behavior of users seeking for information. The resulting corpus consists of three parts:

Part 1. This is the reference part. It corresponds to telephone dialogs between the operator and the caller. It consists of 117 conversations and constitutes the reference group.

Part 2. According to the WOZ technique, the operator simulates a machine with some constraints imposed on the speech production (complete utterances, without hesitations, neither ellipses, nor anaphors, etc.). This part consists of 85 conversations.

Part 3. This is the most restrictive part due to the constraints on the understanding and speech production imposed on the operator who plays the role of a machine. A total of 61 conversations constitute this last part. For parts 2 and 3 of the corpus, a vocoder has been used to fix the fundamental frequency of the operator at a constant level, and to make the user believe to talk to a machine.

***Sundial* Corpus.** Within the framework of the European project SUNDIAL, a spoken language dialog corpus containing air travel information and reservation dialogs via the telephone, has been recorded using the WOZ technique (Bilange, 1992).

The tools used for the dialog simulation contained a standard telephone connected to a tape recorder, a UNIX station with dialog management and database access tools used by the wizard, and also a PC with speech synthesis, connected to the UNIX station. A total of 300 dialogs have been recorded.

***Plus* Corpus.** An analysis of written dialogs for interrogating the yellow pages has been carried out on the basis of corpora. These have been collected within the framework of the European project PLUS (Pragmatic-based language understanding system) (Pernel, 1991; Pernel, 1994).

The dialog corpora, recorded using the WOZ technique, cover three application domains, including personal insurances, car hiring and restaurant information. The database used in this experiment was geographically limited to the city of Paris. For the car rental scenario, agencies at the airport of Roissy-Charles-de-Gaulle have been added.

A total of 36 test persons participated in the experiment, 46 dialogs have been recorded. Three scenarios were proposed to each test person, one for each application domain. The test persons were made believe that they would evaluate a new software intended to replace the one currently available on the French Minitel (videotext) terminal. The latter one is limited to reference words, whereas the new software allowed to understand the majority of written sentences in current French.

3.4.3 Sub-dialog classification

The human-machine dialog is in general divided into sub-dialogs, each yielding a well determined functional value, related to the *task,* the *dialog* or the *meta-dialog.*

Task. The semantic content of these sub-dialogs relates to the task.

Information seeking and response sub-dialogs. The sub-dialog for an information request is opened by the caller requesting some information. This request is equivalent to the main level of the Luzzati model. In the SNCF corpus, these sub-dialogs contain only one request that may be preceded by a formality (*hello*, etc.).

Example III.5
(SNCF Corpus, part 1 – O : Operator, C: Caller)
O : *SNCF bonjour*
(*SNCF good morning*)
C : *oui bonjour, excusez-moi j'aurais besoin d'un renseignement pour faire Paris Génolhac ...*
(*yes good morning, excuse me, I need some information to travel from Paris to Génolhac ...*)

Precision and explanation sub-dialogs. In the analyzed dialogs, the operator asks the caller to be more precise in order to find an appropriate response. The semantic content of the precision requests is, in general, task-specific. It may be related to general information, such as location and time information, but may also be application-specific.

In the first part of the SNCF corpus it can be noted that, whenever the caller requires train schedules, the operator asks him to specify the departure day and time to be able to narrow the search space. It seems that the operator almost always adopts the same strategy to ask for details.

Example III.6
(SNCF Corpus, part 1 – O: Operator, C: Caller)
O : *SNCF bonjour*
(*SNCF good morning*)
C : *bonjour mademoiselle je voudrais avoir les heures de train de Nevers s'il pour Nevers s'il vous plaît*
(*good morning Miss I would like to have the train schedules for Nevers pl.. to Nevers please*)
O : *oui pour quand ?*
(*yes for which date?*)
C : *euh, pour le vingt-six décembre*
(*hmm for December 26th*)
O : *pour le vingt-six décembre ?*
(*for December 26th?*)

***C :** oui*
(*yes*)
***O :** dans la matinée dans l'après ...*
(*in the morning or in the after ...*)
***C :** dans le matin la matinée*
(*in the morning*)

The operator asks a first question on the departure day and a second one on the time range.

In turn, in the second and third parts of the SNCF corpus, the caller, having the impression to talk to a machine, formulates in general much more complete questions, which results in a limited number of precision sub-dialogs.

Contestation and discussion sub-dialogs. After the operator has given a response to the caller, contradiction between his knowledge or beliefs and those of the operator may exist. In order to resolve this problem, a contestation sub-dialog may be opened by the caller. It allows him to express the fact that the system reply corresponds neither to his beliefs nor his expectations.

An interesting example from the SNCF corpus illustrates the surprise of the caller when learning about the traveling time between *Paris* and *Nancy*. He contests this fact with the following sub-dialog:

Example III. 7
(SNCF Corpus, part 1 – O: Operator, C : Caller)
***O :** quatorze heures trente sept, il arrive à dix-huit heures zéro deux*
(*2 37 pm, it arrives at 6 02 pm*)
***C :** oh la la, dix-huit heures zéro deux*
(*oh la la, 6 02 pm*)
***O :** oui à Paris gare de l'Est*
(*yes at Paris eastern train station*)
***C :** il met il met tout* ça ?
(*it takes all this time?*)
***O :** ben oui*
(*oh yes*)

Example III. 8
(SNCF corpus, part 3 – W : Wizard, C : Caller)
***W :** ce train vous convient-il ?*
(*is this train suitable for you?*)
***C :** non on voudrait le suivant s'il vous plaît*
(no we would like the next one please)

Dialog. In this case, the sub-dialogs yield the role of structuring and organizing the dialog.

Formality sub-dialog. Telephone dialogs start, in general, with a sequence of formalities, such as *hello, good morning* or *good evening.* They also end with formalities such as *thanks, good bye.* In the analyzed telephone corpora (SNCF and Air France), the operator starts each dialog with an opening formality *SNCF good morning,* or *Air France good evening.* The caller replies in most cases with *hello* and formulates his information request: *hello Madam, I would like to know the departure time for trains ehm ...* in the same utterance.

However, some callers reply to the formality, but do not continue with formulating their question. This is illustrated by the following example:

Example III. 9
(SNCF Corpus, part 1 – O: Operator, C : Caller)
O : *SNCF* bonjour
(*SNCF good morning)*
C : *allô?*
(*hello?*)
O : *oui*
(yes)
C : *oui bonjour*
(*yes good morning*)
0 : *bonjour*
(*good morning*)
C : *vous pouvez me donner un train pour Châlon, Paris Châlon ...*
(*you may give me a train to Châlon, Paris Châlon ...*)

This formality sub-dialog, taken from a human-human spoken language dialog, contains five turns, which is in contrast to other sub-dialogs containing only two turns.

In the second and third parts of the SNCF corpus of WOZ dialog recordings with a simulated machine the callers tend to directly formulate their question without replying to the formalities. This may be due to the rather direct question *SNCF good morning, which information do you wish?* which incites the caller to be more laconic.

Example III.10

(SNCF Corpus, part 2 – O: Operator, C : Caller)

***O :** SNCF bonjour, quels renseignements désirez-vous ?*

(*SNCF good morning, which type of information do you wish to have?*)

***C :** et ben voilà j'aimerais avoir le prix ...*

(*yes I would like to have the fare ...*)

In this dialog, the caller has replaced the traditional *hello* by the expression *et ben voilà.*

In the PLUS corpus, five closing types have been identified: formal human closing (*good bye*), polite direct closing (*perfect, thank you very much*), laconic direct closing *(we should stop now),* personalized closing *(I have this information already, thank you*) and, finally, immediate stop due to a bad connection to the videotext terminal. In the SNCF corpus, the following closing types have been identified: *good bye, thank you, that's all* and *perfect.*

Here are some illustrative examples:

Example III.11

(SNCF Corpus, part 1 – O : Operator, C : Caller)

***C :** bon ben écoutez je vous remercie*

(*OK. Well, thank you*)

***O :** ben de rien*

(*you are welcome*)

***C :** au revoir*

(*good bye*)

***O :** au revoir*

(*good bye*)

This sub-dialog contains four turns, the closing formality has been initiated by the caller.

Example III.12

(SNCF Corpus, part 1 – O : Operator, C : Caller)

***C :** bon vous êtes bien gentille merci beaucoup*

(*you are very kind, thanks a lot*)

***O :** de rien au revoir*

(*you are welcome, good bye*)

***C :** au revoir*

(*good bye*)

This sub-dialog contains three dialog turns. The operator responds with *you are welcome,* and takes at the same time the initiative to close the dialog by saying *good bye.*

In this sub-dialog, the satisfaction of the caller having obtained the appropriate information is expressed by *you are very* kind and *thanks a lot.*

Example III.13

(SNCF Corpus, part 1 – O : Operator, C : Caller)

C : *bon ben écoutez vous êtes très aimable madame*

(*oh, you are very kind, madam*)

O : *ben merci*

(*oh, thank you*)

C : *merci beaucoup*

(*thank you very much*)

C : *au revoir*

(*good bye*)

O : *au revoir*

(*good bye*)

We note that, in general, all closing sub-dialogs are initiated by the caller.

Meta-dialog. Not directly task or dialog-related sub-dialogs form the meta-dialog. It contains those parts of the discourse that do not contribute to the information gathering process. Meta-dialogs include discussions about the dialog itself, and the management of the communication channel. The latter consists of preserving the contact between both dialog partners (restart, standby), or of evaluating the communication quality.

Resume and standby sub-dialogs. The resume and standby sub-dialogs are symmetrical, the first one yielding the role of restarting the dialog after an interruption, whereas the second one is used to make the caller waiting, e.g., during database access.

The dialog is resumed, when one dialog partner expects an intervention of the other. Typically, in part 2 of the SNCF corpus, the operator starts the dialog with the formality *SNCF good morning,* followed by *which information do you wish?* In part 1, certain callers only formulate their question after having made sure that the communication is well established. In this case, the operator restarts the dialog, which is illustrated by the following example:

Example III.14

(SNCF Corpus, part 1 – O : Operator, C : Caller)

O : *SNCF bonjour – allô ?*

(*SNCF good morning – hello?*)

C : *allô oui*

(*hello, yes*)

O : *oui* (*restarting the dialog with the word oui (yes)*)

C : *est-ce que je pourrais avoir les horaires des trains pour euh Nantes ?* (*may I have the train schedules for ehm Nantes?*)

Correction sub-dialog. Since both, caller and operator are likely to make errors throughout the dialog, they sometimes perform corrections, either by repeating the erroneous part of the utterance, or by replacing it. In the following example from the SNCF corpus, the caller corrects the departure time provided earlier in the dialog.

Example III.15

(SNCF Corpus, part 1 – O : Operator, C : Caller)

C : *non pas quinze heures dix-sept*

(*no, not 3 17 pm*)

O : *ah, dix-sept heures ?*

(*ah, 5 pm?*)

C : *oui*

(*yes*)

Disambiguation sub-dialog. Should the interlocutor be unable to correctly perceive or understand an utterance, or should he detect an unspecified inconsistency, it uses an intervention at the *meta-level* to express his doubts. In case of a total misunderstanding, he may ask his dialog partner to reformulate or completely repeat his utterance. Otherwise, he may ask him to confirm what he has partially heard or understood.

The following example corresponds to a disambiguation sub-dialog containing two sub-dialogs, a repetition (O_2 and C_2) and a confirmation (O_3 and C_3) sub-dialog.

Example III.16

(SNCF Corpus, part 1 – O : Operator, C : Caller)

O$_1$ *: vers quelle heure à peu près vous désirez partir ?*

(*at approximately what time do you want to leave?*)

C$_1$ *: euh huit heures du matin*

(*ehm eight o'clock in the morning*)

O$_2$ *: pardon ?*

(*sorry?*)

C$_2$ *: huit heures du matin ?*

(eight o'clock in the morning?)

O$_3$ *: huit heures du matin*

(eight o'clock in the morning)

C$_3$ *: oui*

(*yes*)

Spoken language dialogs, notably over the telephone mainly consist of disambiguation sub-dialogs, because of an incorrect recognition of certain words or phrases.

Example model. We describe a dialog model (Bennacef et al., 1995; Bennacef et al., 1996; Minker and Bennacef, 2000), that includes the different sub-dialogs identified above.

In fact, in the human-machine spoken language dialog management, it seems appropriate to identify language acts and different sub-dialogs and to organize them hierarchically. This should allow the machine an exhaustive dialog control and an appropriate reply. Thus, the initiative are separated from the reactive parts of the dialog. For example, if an explanation request is formulated by the user as a reaction on a system-generated precision request, it seems appropriate to make the system satisfying the explanation request first, before eventually restarting the precision request. The labeling of specific sub-dialogs allows to keep a trace when certain requests are in standby, and to resume their processing afterwards.

In the following, we present a dialog model that is based jointly on the dialog and language act theory. Similar to other authors (Bilange, 1992; Waterworth, 1982), we use the term *dialog act* to designate elementary components for expressing a co-operative information-seeking dialog (Guyomard and Siroux, 1988; Guyomard et al., 1990). In (Bunt, 1989; Bunt, 1994) the dialog act indicates a function that appeals on the knowledge about both dialog partners, as well as on the dialog structure. The dialog model is viewed as an abstraction of the dialog articulations in the form of dialog acts. Due to this abstraction, the model yields a general control on the dialog flow.

In contrast to the models based on linguistic studies the presented dialog model is non-descriptive. It corresponds to a functional model aiming to control a spoken language dialog system. This latter one interprets the user utterance, dynamically chooses between different dialog strategies, and adapts the system feedback to the dialog context. Furthermore, the dialog system needs to guide the user to carry out a simple, concise and effective dialog and to enable anticipation/prediction throughout the dialog.

A dialog act is composed of prepositional content and an illocutionary function. The prepositional content corresponds, in our example, to the frame generated by the case grammar analyzer (cf. chapter 2). The illocutionary function corresponds to the function played by the dialog act in the dialog process. In the following, we classify the dialog acts by their illocutionary function (e.g., act of precision request, etc.).

The dialog model is based on a formal grammar equivalent to the one used in linguistic analysis. The non-terminals of the grammar correspond to the sub-dialogs, whereas the terminals correspond to the dialog acts. According to this definition, the sub-dialog concept needs to be apprehended carefully since it may correspond to only one dialog act. In fact, the presented model corresponds to the hierarchical and functional principles of the Geneva model.

We conclude from the corpora studies that the dialog is chronologically built up of three phases, each which contains one or several sub-dialogs:

Opening formality. It corresponds to the formalities of the starting dialog.

Information. In this phase, different exchanges between both interlocutors allow to obtain information. This phase represents the most significant part of the dialogs of information request and consequently constitutes, the core part of the dialog analysis.

Closing formality. It concerns the formalities that terminate the dialog.

The dialog consists of three major sub-dialogs, an information sub-dialog *SDInf,* preceded and eventually followed by opening and closing formality sub-dialogs *SDForO* and *SDForF* respectively. This configuration results in the following equation, where the sub-dialog represented in bold is mandatory:

dialog = opening formality + ***information***
+ closing formality

or by the rewriting rule :

$$D = \{SDForO\}^* \; \{SDInf\}^+ \; \{SDForF\}^* \tag{3.1}$$

The symbols between accodances $\{\}$ may be followed by the $*$ or $+$ signs. A non-terminal followed by $*$ occurs zero or an infinite times, whereas, if followed by +, it occurs at least once. The slash / in the right rule members stands for an *exclusive or.*

We now examine each of the three sub-dialogs.

Opening formality. It may consist of one or several opening formality acts *ForO,* resulting in the rule:

$$\{SDForO\} = ForO^* \tag{3.2}$$

The application of this rule is illustrated by the following example:

Example III.17

(SNCF Corpus – O : Operator, C : Caller)

O : *SNCF bonjour* (***ForO***)
(*SNCF good morning*)
C : *allô ?* (***ForO***)
(*hello?*)
O : *oui* (***ForO***)
(*yes*)
C : *oui bonjour* (***ForO***)
(*yes good morning*)

$(\boldsymbol{SDForO})$

Information. This sub-dialog, the most significant part of a dialog for information request, is primarily made up of an information request sub-dialog *SDDInf.* This latter one corresponds to an information request act, i.e., to an information request, eventually preceded by a resumption *SDRes,* and followed by a precision sub-dialog *SDPr. SDRes* is used to resume the dialog, *SDPr* is triggered should some precision be necessary for the database access and information retrieval. The response corresponding to the information request act *R* may be preceded by a standby sub-dialog *SDAt* used to maintain the communication. It may be followed by the contestation and discussion sub-dialogs *SDCont* and *SDDisc* respectively.

To a certain extent *SDInf* may be broken down to an information request and a response to this request. This sub-dialog is formalized by the following equation (the parts represented in bold are mandatory):

information = restart + information request + precision
+ standby + ***response*** *+ contestation + discussion*

or by the rewriting rule :

$$\begin{aligned} \{SDInf\} \;=\; & \{SDRel\}^* \, \{SDDInf\}^* \, \{SDPr\}^* \\ & \{SDAt\}^* \, R \, \{SDCont\}^* \, \{SDDisc\}^* \end{aligned} \tag{3.3}$$

We now examine each part of the information sub-dialog.

Restart sub-dialog. It may be built by one or more acts of resumption *SDRes* and formalized by the rule:

$$\{SDRes\} = Res^* \tag{3.4}$$

Example III.18

(Corpus SNCF – O : Operator, C : Caller)

O : *SNCF bonjour (**ForO**)*
(SNCF good morning)
C : *allô oui ? (**ForO**)*
(hello yes?)
$(\boldsymbol{SDForO})$

O : *je vous écoute (**Rel**)*
(how may I help you?)
C : *oui, ... (R)*
(yes, ...)
$(\boldsymbol{SDRel})$

Even though this act may occur any time in the dialog, it seems most appropriate to plan it at the beginning of the information sub-dialog.

Information request sub-dialog. This sub-dialog, formalized by the following rule, corresponds to an information request act *DInf*:

$$\{SDDinf\} = DInf \tag{3.5}$$

Example III.19

(SNCF Corpus – O: Operator, C : Caller)

O : *SNCF bonjour (**ForO**)*
(SNCF good morning)
$(\boldsymbol{SDForO})$

C : *oui, j'aurais besoin d'un renseignement pour faire*
*Paris Génolhac... (**Dinf**)*
(yes I need some information to go from Paris to Génolhac ...)
$(\boldsymbol{SDDInf})$

Precision and explanation sub-dialogs. A precision sub-dialog *SDPr* corresponds to a precision request act *DPr* claimed by the operator *O* on some parameters of the information request. It also corresponds to an explanation sub-dialog *SDExp* opened by the caller *C,* and to a response act *R* to *DPr.* In turn, the *SDExp* sub-dialog contains an explanation request act *DExp,* a *SDPr* sub-dialog, and a response act *R* to *Dexp.* In fact, there exists a recursive definition of each sub-dialog: *SDPr* is defined on the basis of *SDExp,* which in turn is defined based on *SDPr,* as this may be inferred from the following rules:

$$\begin{array}{lcl} \{SDPr\} & = & \{SDPr\}\ \{SDExp\}^*\ R \\ \{SDExp\} & = & \{SDExp\}\ \{SDPr\}^*\ R \end{array} \tag{3.6}$$

Both sub-dialogs may be compared with the incidental sub-dialogs of the Luzzati model. They express the fact that before replying to the precision requests from the operator *O*, the caller *C* may ask for some explanation about these precisions. Similarly, *O* may ask *C* to provide some details about his request for explanation.

For the model implementation described later, the recursivity has been limited to only two levels.

Example 111.20

(PLUS Corpus – *O : Operator, C : Caller)*
***C :** souhaitez-vous une mutuelle ou une assurance*
*normale ? (**DP**r)*
(*do you wish a special or normal insurance?*)
***O :** je ne sais pas quelles sont les différences ? (**DExp**)*
(*I don't know the difference?*) $(SDExp)$
***C :** les tarifs des mutuelles sont en général moins chers*
*car elles choisissent leurs clients (**R**)*
(*the tarifs of the special insurances are in general less expensive*
because they choose their clients) $(SDPr)$

***O :** mais du point de vue de la couverture ? (**DExp**)*
(*but from the point of view of the coverage?*)
***C :** cela dépend des contrats, mais vous devez pouvoir*
*trouver les mêmes couvertures (**R**)* $(SDExp)$
(*this depends on the contract but you should*
find the same coverage)
***O :** je souhaite être complètement couverte ... (**R**)*
(*I would like to be entirely covered ...*)

Standby sub-dialog. Should the information retrieval require a certain time, a standby sub-dialog *SDSt* is opened by a standby act *St.* This sub-dialog is formalized by:

$$\{SDSt\} = St^* \tag{3.7}$$

Example III.21

(SNCF Corpus – O : Operator, C : Caller)
***O :** ah, ne quittez pas s'il vous plaît (**St**)*
(*ah, hold on please*) $(SDSt)$
***C :** oui merci (**St**)*
(*yes thank you*)

A standby act may appear any time in the dialog. However, a particular role should be attributed to the standby acts in applications for information requests. In fact, in these applications, the standby is frequently observed when information is retrieved from the database.

Response sub-dialog. It corresponds to a response act *R* to the information request *DInf* :

$$\{SDR\} = R \tag{3.8}$$

<u>*Example III.22*</u>

(SNCF Corpus – O : Operator, C : Caller)

C : *est-ce que je pourrais avoir les horaires des trains pour euh Nantes ? (**DInf**)* (*may I get the train schedules for ehm Nantes?*)	$(\boldsymbol{SDDInf})$
O : *bon alors vous avez un train à seize heures trente à Paris Montparnasse (**R**)* (*ok you have a train from Paris Montparnasse at 4 30 pm*)	$(\boldsymbol{SDR})$

Contestation sub-dialog. After the system feedback generation according to the information request act, one or several contestation sub-dialogs *SDCont* may be opened by the information-seeking caller.

These sub-dialogs are formalized by the following rule, which has the contestation act *Cont* as a right member, followed by a response act *R* referring to this contestation:

$$\{SDCont\} = Cont\ R \tag{3.9}$$

<u>*Example III.23*</u>

(SNCF Corpus – O : Operator, C : Caller)

O : *quatorze heures trente sept, il arrive à dix huit heures zéro deux (**R**)* (*2 37 pm, it arrives at 6 02 pm*) ***C :*** *oh la la, dix huit heures zéro deux ? (**Cont**)* (*oh la la, 6 02 pm?*) ***O :*** *oui à Paris gare de l'Est (**R**)* (*yes at Paris eastern train station*)	$(\boldsymbol{SDCont})$	$(\boldsymbol{SDCont})$
C : *il met il met tout ça ? (**Cont**)* *(it needs all that time?)* ***O :*** *ben oui (**R**)* (*hmm yes*)	$(\boldsymbol{SDCont})$	

Discussion sub-dialog. A contestation may be followed by a discussion sub-dialog, in which the interlocutor having asked for information initiates a discussion about the feedback of the system.

A discussion sub-dialog, formalized by the following rule, corresponds to a discussion act *Disc,* followed by a reply act *R*:

$$\{SDDisc\} = Disc\ R \tag{3.10}$$

Example III.24

(SNCF Corpus – O : Operator, C : Caller)
***O :** l'adresse et le téléphone de AVIS sont : ... (**R**)*
(the address and telephone number of AVIS are: ...)
***C :** y a que AVIS ? (**Disc**)*
(is there only AVIS?)
***O :** non, il y a aussi EUROPCAR (**R**)*
(no there is EUROPCAR as well)

$(\boldsymbol{SDDisc})$

***C :** quelle compagnie me conseillez-vous ? (**Disc**)*
(which company do you recommend?)
***O :** je ne peux pas vous répondre, je n'ai pas ce type de renseignement (**R**)*
(I cannot reply, I do not have this type of information)

$(\boldsymbol{SDDisc})$

$(\boldsymbol{SDDisc})$

Closing formality. The end of the dialog is generally characterized by closing formality sub-dialogs *SDF or C.* Similar to the opening formality sub-dialog, it is built by one or several closing formality acts *ForC*:

$$\{SDForC\} = ForC^{*} \tag{3.11}$$

Example III.25

(SNCF Corpus – O : Operator, C : Caller)
***C :** bon ben écoutez vous êtes très aimable madame (**ForC**)*
(well, you are very kind Madam)
***O :** oh ben merci (**ForC**)*
(oh well, thank you)
***C :** merci beaucoup (**ForC**)*
(thanks a lot)
***C :** au revoir (**ForC**)*
(good bye)
***O :** au revoir (**ForC**)*
(good bye)

$(\boldsymbol{SDForC})$

Correction and disambiguation sub-dialogs. The correction and disambiguation sub-dialogs may be appended to each sub-dialog presented above. Using these sub-dialogs the interlocutor may wish to correct himselves *SDCor* or to disambiguate *SDDis* in case of an utterance, that has been badly understood by the system. The corresponding rule is:

$$\{SDCD\} = \{SDCor\}^{*}\ /\ \{SDDis\}^{*} \tag{3.12}$$

Correction sub-dialog. It corresponds to one or more correction acts *Cor* and is formalized by:

$$\{SDCor\} = Cor^* \tag{3.13}$$

Example III.26

(SNCF Corpus – C : Caller)
***C :** non pas quinze heures dix sept (**Cor**)*
(not 3 17 pm)
$(\boldsymbol{SDCor})$

Disambiguation sub-dialog. It may contain sub-dialogs for reformulation *SDRef* or confirmation *SDConf*:

$$\{SDDes\} = \{SDRef\}^* \ / \ \{SDConf\}^* \tag{3.14}$$

SDRef contains a reformulation request act *DRef,* followed by a reply act *R. SDConf* contains a confirmation request act *DConf,* followed by a reply act *R.* The rules corresponding to both sub-dialogs are formalized by:

$$\begin{array}{lcl} \{SDRef\} & = & DRef\ R \\ \{SDConf\} & = & DConf\ R \end{array} \tag{3.15}$$

Example III.27

(SNCF Corpus – O: Operator, C : Caller)
***O :** vers quelle heure à peu près vous désirez partir ? (**DPr**)*
(about what time do you want to leave?)
***C :** euh huit heures du matin (**R**)*
(euh eight o'clock in the morning)
***O :** pardon ? (**DRef**)*
(sorry?)
***C :** huit heures du matin ? (**R**)*
(eight o'clock in the morning?)
$(\boldsymbol{SDRref})$

***O :** huit heures du matin (**DConf**)*
(eight o'clock in the morning)
***C :** oui (**R**)*
(yes)
$(\boldsymbol{SDConf})$

$(\boldsymbol{SDPr})$

Model restrictions. The presented dialog model should be able to appropriately model a spoken language dialog between a user and a machine. The user may be compared to the caller in the presented examples. The machine plays the role of the operator or the wizard.

In a human-machine spoken language dialog, the machine starts the dialog by *SDForO* formalities. The user responds. Eventually after being re-initiated by a machine-generated *SDRes* the user starts an information request sub-dialog by posing a *SDDInf* question. Should this question be incomplete with respect to the task, the machine asks for details *SDPr.* The user may provide these details, or ask for some explanations *SDExp.* The machine may require further precisions on these explanations, and this even recursively until the user has provided an appropriate feedback. During database access, the machine may open a standby sub-dialog *SDSt* before generating a response *SDR* to the principal user request. Once this response provided to the user, he may contest it *(SDCont),* and/or discuss it *(SDDisc).* Finally, and on the initiative of the user, the dialog is terminated by closing formalities *SDForC.*

In the described approach, the sub-dialogs are not symmetrical. Depending on the task properties, their usage may depend on whether they are generated by the user, the machine, or by both:

Category 1. It represents the user-generated acts *DInf, DExp, Cont, Disc* and *Cor.* In fact, the user asks for information by using the *DInf* act, requires an explanation by *DExp,* contests and discusses a reply provided by the machine by *Cont* and *Disc* respectively, and corrects himself during the *Cor* dialog.

Category 2. This category contains the system-specific acts including *Res* to resume the dialog, *St* to set the user on standby during the information retrieval and, finally, the precision request *DPr* to complete the user query.

Category 3. The acts generated either by the user or the machine are contained in this category. These are *ForO, R, Dref, Dconf* and *ForF.* The response act *R* corresponds to a reaction to the last identified act which differs from *R.* Therefore, it seems reasonable to use only one type of response acts. It may be exhaustively described in the course of the dialog process according to the last identified act.

Before presenting the complete dialog model, we show how the correction and disambiguation sub-dialogs *SDCD* may fit into other sub-dialogs.

Theoretically, using *SDCD* may be appropriate each time a user generates an utterance. However, there exist a number of restrictions in a human-machine spoken language dialog:

Unlike humans, the machine never performs self-corrections. It generates pre-formulated utterances, without any hesitations nor self-corrections. Thus, any correction sub-dialogs following machine-generated dialog acts do not exist. Only reformulation requests *SDRef* may follow *ForO, ForC, Res* and *St.* Users almost never correct themselves nor ask in general for a confirmation of these types of utterances. A system asking *did you really say hello?* would seem rather strange.

We are now able to classify the sub-dialogs according to the three main categories, *task*, *dialog* and *meta-dialog.*

Task. The contents of the sub-dialogs *SDDInf, SDPr, SDExp, SDCont, SDDisc* are task-related.

Dialog. The sub-dialogs *SDForO* and *SDForC* provide structure and organize the dialog.

Meta-dialog. The sub-dialogs *SDSt, SDCor, SDDis* and *SDRel* manage the communication channel and depend on what has been said during the dialog.

3.4.4 Dialog Model Formalization

After the detailed analysis of each sub-dialog, we are now able to present a complete formalized dialog model, in the form of a grammar that is similar to those used in language theory. We have accounted for the above-mentioned restrictions to introduce the sub-dialogs *SDCD, SDDis* and *SDRef.*

G is defined as the dialog grammar by the quadruplet:

$$G(V_n, V_t, R, D) \tag{3.16}$$

with:

V_n: non terminal vocabulary containing the different identified sub-dialogs, described in Table 3.1,

V_n = {*SDForO, SDInf, SDForC, SDRes, SDDInf, SDPr, SDExp, SDSt, SDR, SDCont, SDDisc, SDSC, SDCor, SDDis, SDRef, SDConf*},

V_t: terminal vocabulary corresponding to the dialog acts described in Table 3.2,

V_t = {*ForO, Res, DInf, DExp, St, R, Cont, Disc, Cor, DRef, DConf, ForF*},

R: rewriting rules (cf. Figure 3.6),

D: grammar axiom or start symbol.

The complete formalized dialog model is presented in Figure 3.6.

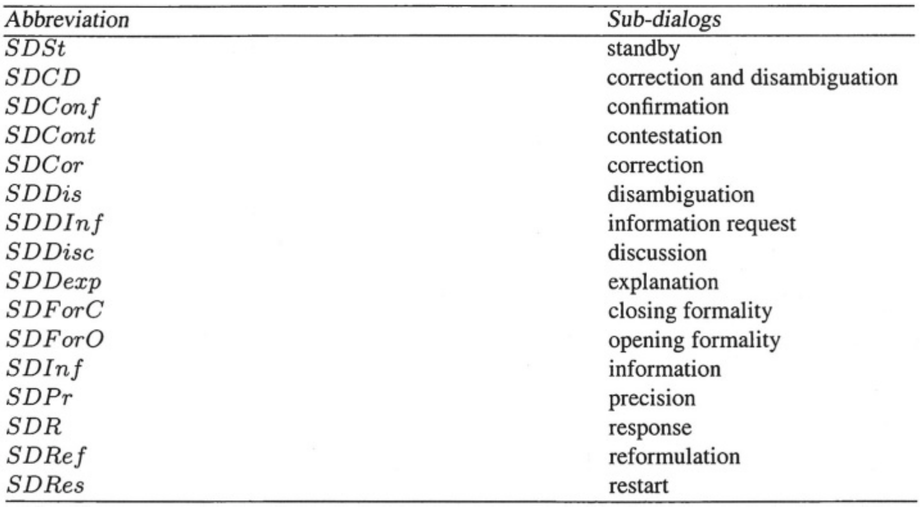
3.4.5 Dialog Model Representation

The dialog model that corresponds to our example is represented by a deterministic finite-state machine. It provides a powerful implementation model. The grammar related to the dialog model is context-free. It is not equivalent to an automaton or a regular grammar. However, the complete set of grammar rules, except those for precision and explanation sub-dialogs (due to their recursivity), may be transformed (Aho and Ullman, 1972; Greibach, 1965) into regular rules of type:

$X \rightarrow aY$, with *a* as a terminal, and *X, Y* as non-terminals.

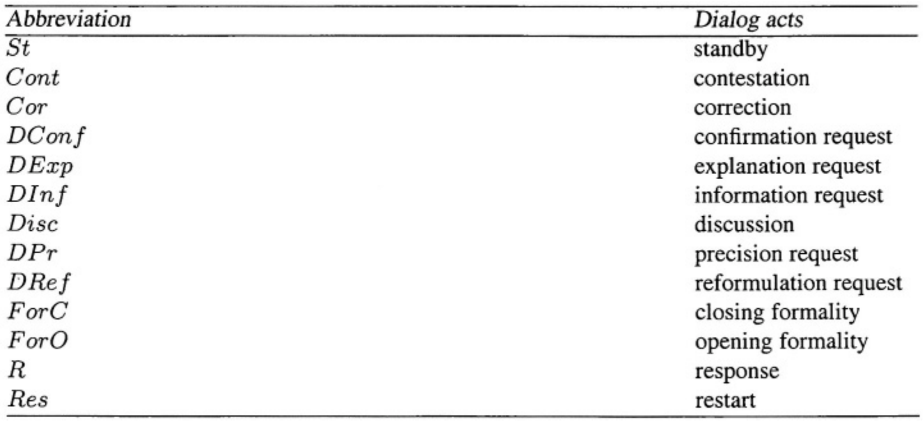
For both rules, an infinite recursivity for precision and explanation sub-dialogs is not required, because in the analyzed corpora, a precision request has never occurred in an explanation sub-dialog. Nevertheless, with the aim of

Table 3.1. **Sub-dialogs in the formalized dialog model.**



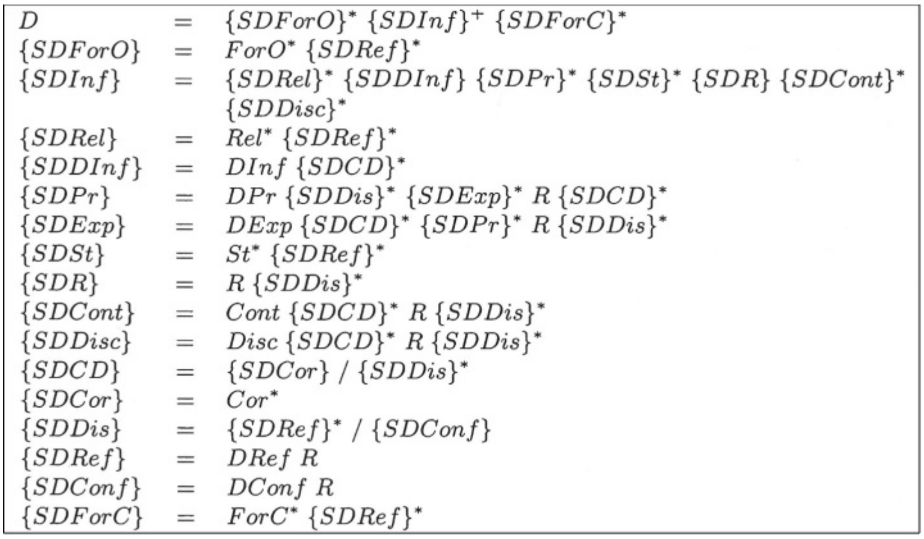
Abbreviation	*Sub-dialogs*
$SDSt$	standby
$SDCD$	correction and disambiguation
$SDConf$	confirmation
$SDCont$	contestation
$SDCor$	correction
$SDDis$	disambiguation
$SDDInf$	information request
$SDDisc$	discussion
$SDDexp$	explanation
$SDForC$	closing formality
$SDForO$	opening formality
$SDInf$	information
$SDPr$	precision
SDR	response
$SDRef$	reformulation
$SDRes$	restart

Table 3.2. **Dialog acts in the formalized dialog model.**



Abbreviation	*Dialog acts*
St	standby
$Cont$	contestation
Cor	correction
$DConf$	confirmation request
$DExp$	explanation request
$DInf$	information request
$Disc$	discussion
DPr	precision request
$DRef$	reformulation request
$ForC$	closing formality
$ForO$	opening formality
R	response
Res	restart

a model generalization, two recursivity levels should be enabled by the following rule transformation:



$$
\begin{array}{lcl}
D & = & \{SDForO\}^* \ \{SDInf\}^+ \ \{SDForC\}^* \\
\{SDForO\} & = & ForO^* \ \{SDRef\}^* \\
\{SDInf\} & = & \{SDRel\}^* \ \{SDDInf\} \ \{SDPr\}^* \ \{SDSt\}^* \ \{SDR\} \ \{SDCont\}^* \\
 & & \{SDDisc\}^* \\
\{SDRel\} & = & Rel^* \ \{SDRef\}^* \\
\{SDDInf\} & = & DInf \ \{SDCD\}^* \\
\{SDPr\} & = & DPr \ \{SDDis\}^* \ \{SDExp\}^* \ R \ \{SDCD\}^* \\
\{SDExp\} & = & DExp \ \{SDCD\}^* \ \{SDPr\}^* \ R \ \{SDDis\}^* \\
\{SDSt\} & = & St^* \ \{SDRef\}^* \\
\{SDR\} & = & R \ \{SDDis\}^* \\
\{SDCont\} & = & Cont \ \{SDCD\}^* \ R \ \{SDDis\}^* \\
\{SDDisc\} & = & Disc \ \{SDCD\}^* \ R \ \{SDDis\}^* \\
\{SDCD\} & = & \{SDCor\} \ / \ \{SDDis\}^* \\
\{SDCor\} & = & Cor^* \\
\{SDDis\} & = & \{SDRef\}^* \ / \ \{SDConf\} \\
\{SDRef\} & = & DRef \ R \\
\{SDConf\} & = & DConf \ R \\
\{SDForC\} & = & ForC^* \ \{SDRef\}^*
\end{array}
$$

Figure 3.6. Rewriting rules in the complete formalized dialog model.

$$
\begin{array}{lcl}
\{SDPr1\} & = & DPr \ \{SDDis\}^* \ \{SDExp1\}^* \ R \ \{SDCCD\}^* \\
\{SDExp1\} & = & DExp \ \{SDCD\}^* \ \{SDPr2\}^* \ R \ \{SDDes\}^* \\
\{SDPr2\} & = & DPr \ \{SDDis\}^* \ \{SDExp2\}^* \ R \ \{SDCD\}^* \\
\{SDExp2\} & = & DExp \ \{SDCD\}^* \ R \ \{SDDis\}^*
\end{array}
\qquad (3.17)
$$

3.4.6 Dialog Analysis Example

We illustrate the complete dialog model on a real example dialog from the SNCF corpus.

<u>*Example III.28*</u>

(SNCF Corpus, part 1 – O : Operator, C : Caller)

O : *SNCF bonjour* (***ForO***)
(*SNCF good morning*)
allô ? (***ForO***)
(*hello?*)
C : *allô oui ?* (***ForO***)
(*hello yes?*)

$(\boldsymbol{SDForO})$

O : *oui* (***Res***)
(*yes*)

$(\boldsymbol{SDRes})$

C : *est-ce que je pourrais avoir les horaires des trains pour euh Nantes ?* (***DInf***)
(*may I have the train schedules to ehm Nantes?*)

$(\boldsymbol{SDDInf})$

O : *pour Nantes quel jour ?* (***DPr***)
(*to Nantes which day?*)
C : *euh vendredi soir* (***R***)
(*ehm Friday evening*)
O : *euh vendredi qui vient ?* (***DConf***)
(*ehm next Friday?*)
C : *oui vendredi qui vient* (***R***)
(*yes next Friday*)

$(\boldsymbol{SDConf})$

$(\boldsymbol{SDPr})$

O : *oui, ne quittez pas s'il vous plaît* (***St***)
(*yes, please standby*)
C : *oui* (***St***)
(*yes*)
O : *allô ?* (***St***)
(*hello?*)
C : *oui* (***St***)
(*yes*)

$(\boldsymbol{SDSt})$

O : *bon alors vous avez un train à seize heures trente à Paris Montparnasse* (***R***)
(*ok you have a train at 4 30 pm to Paris Montparnasse*)

$(\boldsymbol{SDR})$

C : *ouais je vous remercie au revoir* (***ForC***)
(*yes, thank you and good bye*)
O : *au revoir* (***ForC***)
(*good bye*)

$(\boldsymbol{SDForC})$

3.5 Discussion

As the result of the linguistic research at Geneva University a hierarchical and functional dialog model has been proposed. It represents a theoretical framework for dialog analysis. The Geneva model has been adapted by some state-of-the-art systems for operational purposes and served as a basis for certain parts of the described work.

Nevertheless, there still exists a separation between the structural component of the model in terms of exchange, intervention and language act on the one hand, and the functional component in terms of illocutionary and interactive functions on the other. In fact, the grammar rules of the different models are not sufficient to direct and to control the dialog, as already mentioned in (Bilange, 1992):

> *The nature of the rules for control and conversation differs from the grammar rules of the dialog. In fact, the latter ones, if they express structural constraints, only show the possible dialog flows, whereas the former two categories express the actions directly. If we analyze the presented model, the grammar rules thus express constraints of illocutionary sequences while the two other categories are much closer to the dialog acts since they imply the realization of particular actions.*

We have described a dialog model that combines the structural and functional components and enables a refined dialog control. It includes several sub-dialogs, which may be aligned on three axes, the task, the dialog and the meta-dialog. Compared to the hierarchy, i.e., exchange, intervention and language act, the sub-dialogs separate the different dialog phases and therefore enable a rather precise control of the dialog process. In this work, the sub-dialogs and dialog acts have been determined on the basis of corpora analyses, not from a linguistic point of view, but with the aim to offer a functional model that may directly be used for a given type of applications.

We may consider the dialog act to be composed of a propositional content and an illocutionary function. In the presented model, the propositional content corresponds to the frame generated by the semantic case grammar analyzer. The illocutionary function corresponds to the function played by the dialog act in the dialog process. For example, a dialog act whose propositional content is related to the request for one or several task parameters, plays the role of a precision request in the dialog. Dividing an utterance into dialog acts generally corresponds to the segmentation of concepts carried out by the case grammar analyzer.

It should be noted, that the response act constitutes the generic class of the reactives. It is interpreted with respect to the preceding act. For example, a response act, generated as a result of a precision request, corresponds to a precision.

The presented dialog model contains several sub-dialogs. In the human-machine spoken language dialog management, it seems indeed appropriate to identify dialog acts and different sub-dialogs, and to organize them hierarchically. This enables a precise dialog control and allows the system to provide relevant feedback to the user. Thus, initiative are separated from reactive parts of the dialog. For example, if after a precision request generated by the machine, an explanation request is formulated by the user, it seems appropriate that the machine satisfies this explanation request, before eventually restarting the precision request.

The described model indicates the dialog state after a dialog act identification. This process allows to identify the functions played by each user utterance in the dialog context.

Anticipation/prediction is possible during the dialog on the basis of dialog acts. This prediction may be propagated down to the lower language levels, such as speech recognition and understanding, by associating the possible surface form to each act.

Finally, an explicit information about the task content does not exist. The dialog model yields the advantage of being usable in different information-seeking applications, and may be extended to other types of applications, by adding, for example, specific sub-dialogs of argumentation and explanation.

4. Dialog System Example

In the remainder of this chapter, we describe an example of a spoken language dialog system using a speaker-independent speech recognizer. First we show how to implement the different models that are necessary for the human-machine spoken language dialog. Then we explain the management of semantic representations including the history, which seems to be essential in this type of systems. We describe methods to identify and to generate dialog acts, as well as to manage sub-dialogs and over-information. Global algorithms explain the general systems operations, but focus is placed on the collaboration between the task and dialog models. Finally, we address a methodology for spoken language dialog systems design.

4.1 Architecture

Studies on human-machine spoken language dialog have shown the diversity and complexity of the knowledge required by a dialog system. We have discussed this knowledge in the introduction of this book: morphological, syntactic, semantic, pragmatic and contextual knowledge, knowledge about the dialog, the task to be realized, the dialog partner, and even about the system itself. This knowledge may be represented in a procedural or declarative way, with a more or less artificial separation between them. To each type of knowledge correspond modules that perform a specific processing, including speech recognition, morphological analysis, syntactic-semantic analysis, dialog management mechanisms, etc. Different modular spoken language dialog systems architectures may be distinguished mainly as a function of the communication modes that exist between these modules.

The example dialog system described below is characterized by different sub-processes that are managed by a central dialog process, called the *dialog manager.* It activates or deactivates these underlying processes. Each process operates on a knowledge base with the aim to transform the input information into a specific representation suitable for accessing the application back-end. The processes subsequent to the speech recognition component enable the storage of output representations in a short term memory called the *dialog context.* The representations are stored as networks of frames. The communication between the different processes is performed by means of the context, where each process introduces and withdraws information. The dialog manager thus yields the role of controlling the different processes by activating or deactivating them depending on the *context state*. The context contains the semantic representation of the current utterance, the dialog history, as well as the dialog state.

Figure 3.7 shows the architecture of the dialog system, presented in the introduction of this book (cf. Figure 1.1). The user generates an utterance, which is recognized by the speech recognition component in the form of a word sequence, and then processed by the semantic analyzer (cf. chapter 2). This analyzer, depending on the syntactic and semantic knowledge contained in the case grammar, generates the semantic representation of the user utterance in the form of a network of frames, which is stored in the dialog context. On the basis of this network, the task and the dialog model, other processes in the dialog management module are activated to establish a dialog, to send a command to the application back-end (DBMS) and to generate a feedback to the user.

We are now describing the dialog module in more detail (cf. Figure 3.8):

- On the basis of the network of frames, output of the semantic analyzer, and on the basis of the dialog state, the dialog act corresponding to the first frame (current frame) of the user utterance (dialog context) is identified, to initiate a change of the dialog state (module 1).

- If the dialog act is associated to the dialog or the meta-dialog, a new generated act based on the dialog model and state, is translated into a surface form. This is the inverse of the identification process (module 2).

- For task-related dialog acts, the current frame is completed by the result of the ruled-based interpretation and by the information contained in the dialog history (module 3). The inconsistency detection rules are activated

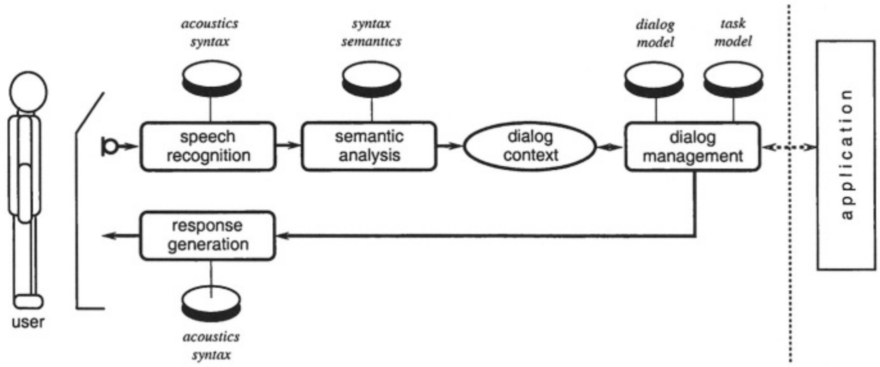


Figure 3.7. Detailed spoken language dialog systems architecture.

to allow the system to ask the user for confirmation (module 5). If the information, necessary to perform the task, cannot be gathered, a precision request is generated on the basis of the plans defined in the task model (module 5). This is done until the totality of the required information is available to allow a command generation towards the application back-end (module 4). With the system response it is possible to generate a feedback to the user (module 5). To each system-generated message corresponds one dialog act.

The advantage of the described architecture relies in the fact that the processes are not sequentially triggered. They are instantiated by the dialog manager yielding the entire knowledge of the system status and, consequently, a good control strategy for the totality of the dialog processes. Furthermore, this architecture seems rather flexible, since the release of the processes is not determined in advance, but depends on the dialog context. Various release strategies may be considered.

4.1.1 General Algorithm

The general algorithm of the dialog manager is described in Figure 3.9.

It is divided into two phases. The system first opens the dialog by generating an opening formality message. The corresponding act *ForO* is used to progress in the dialog automaton.

The second phase corresponds to the main loop of the dialog. Each user utterance is transformed by the semantic analyzer into a succession of frames. If the identified dialog act is related to the dialog or the meta-dialog a new act

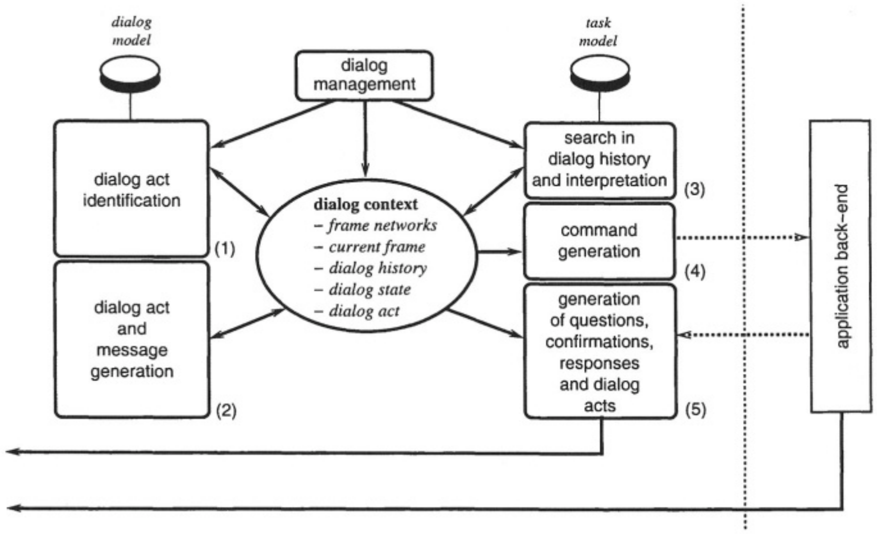


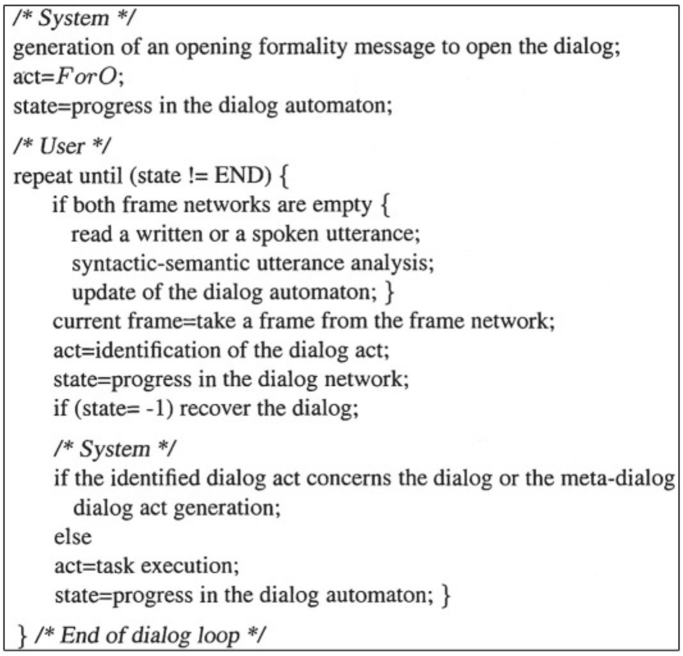
Figure 3.8. Functional architecture of the dialog module.

is generated by the system. Otherwise, the task model processes are started according to the algorithm presented in Figure 3.10.

After completion of the current frame, the ellipsis resolution is activated and the interpretation rule are released. Missing mandatory attributes are replaced by default values. For example, if the departure day has not been specified in the utterance and does not exist in the dialog history, the current day is taken by default. The interpretation rules yield priority on the dialog history, since they access information contained in the utterance. If, for example, the user asks *I want to leave this morning,* the system should infer that it is the current day without having to look up in the dialog history. On the other hand, if the day is not specified in the utterance but contained in the dialog history, the latter one should be considered instead of calculating default values. It should be noted that the interpretation rules and the rules for default value calculation only instantiate empty or non-existing attributes.

After processing the current frame, the coherence rules are activated to ask the user for confirmation in the case of inconsistency. Precision requests are then generated on the basis of model and task plans.

If the complete information necessary to perform a particular task is gathered, a command is sent to the application back-end. Finally, a system feedback is generated to the user.



```
/* System */
generation of an opening formality message to open the dialog;
act=$ForO$;
state=progress in the dialog automaton;

/* User */
repeat until (state != END) {
    if both frame networks are empty {
      read a written or a spoken utterance;
      syntactic-semantic utterance analysis;
      update of the dialog automaton; }
    current frame=take a frame from the frame network;
    act=identification of the dialog act;
    state=progress in the dialog network;
    if (state= -1) recover the dialog;

    /* System */
    if the identified dialog act concerns the dialog or the meta-dialog
      dialog act generation;
    else
    act=task execution;
    state=progress in the dialog automaton; }
} /* End of dialog loop */
```

Figure 3.9. General algorithm of the dialog manager.

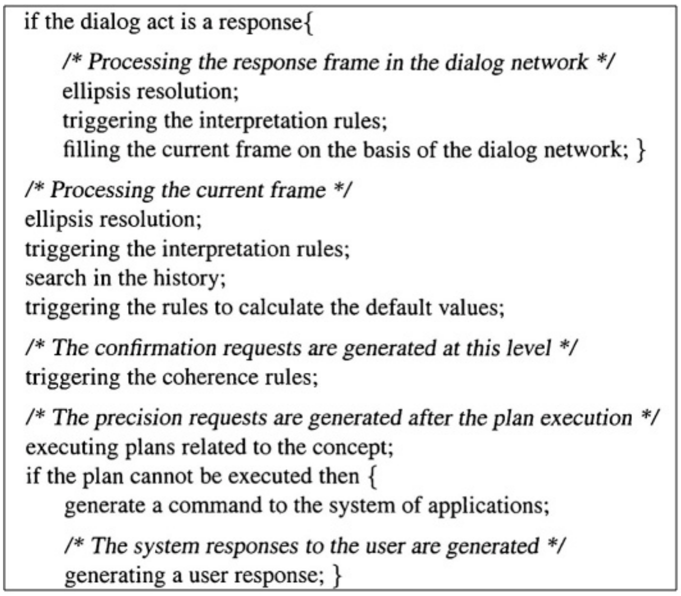
4.2 Utterance Generation

The utterance generation is performed in two steps, a conceptual generation at a semantic level, and a surface generation at a syntactic and lexical level. The conceptual generation defines *what to be said*, whereas the surface generation the way of *how to say it.*

The conceptual representation is obviously language-independent. It is well suited to operations such as history backup generation. At each dialog stage, the dialog manager may look up the previously generated concepts. The conceptual structures are generated by rules that are similar to those described in the task model.

The surface generation is language-dependent. It also depends on the style of language used by the system. Rules choose the most relevant lexical elements that may be used to build a syntactically correct utterance.

It seems important that the system yields identical linguistic abilities in both language analysis and generation.



```
if the dialog act is a response{
    /* Processing the response frame in the dialog network */
    ellipsis resolution;
    triggering the interpretation rules;
    filling the current frame on the basis of the dialog network; }
/* Processing the current frame */
ellipsis resolution;
triggering the interpretation rules;
search in the history;
triggering the rules to calculate the default values;
/* The confirmation requests are generated at this level */
triggering the coherence rules;
/* The precision requests are generated after the plan execution */
executing plans related to the concept;
if the plan cannot be executed then {
    generate a command to the system of applications;
    /* The system responses to the user are generated */
    generating a user response; }
```

Figure 3.10. Task execution algorithm.

4.3 Discussion

We have described an implementation of a dialog manager that integrates the task and the dialog model.

The dialog model is realized as a finite state automaton whose states correspond to those of the dialog and whose transitions to the language acts. The language act is identified on the basis of the utterance, of its semantic content, the current dialog state, the last recognized language act, as well as the dialog history. The language acts may be classified according to whether they are task-dependent (requests for information, precision, explanation, contestation and discussion), whether they are related to the dialog (opening and closing formalities), or to the meta-dialog (reformulation, confirmation, correction, standby and restart). The illustrated dialog model enables a good control of the dialog flow and the system to identify user utterances, to adapt the system behavior, and to generate the most relevant system feedback according to the dialog state.

We have demonstrated how certain simple phenomena of ellipses may be resolved by using the dialog history. We have generally noticed that ellipses in the dialog refer to the preceding utterance and may thus be resolved by using the dialog history.

We have also discussed methods used for dialog act identification, as well as for sub-dialog management. The dialog act identification is performed using the propositional content (frame), certain linguistic surface markers (interrogative adjectives), the dialog state, the last identified dialog act as well as the dialog history. However, these rules still remain insufficient. It would be necessary to further analyze recorded dialog corpora, in order to identify the totality of the relevant features that are able to contribute to the dialog act recognition.

5. Conclusion

In this chapter we have discussed aspects for modeling human-machine spoken language dialog.

Several systems have been adapted the Geneva model for operational purposes. Nevertheless, there always exists a separation between the structural model component in terms of exchange, intervention and language acts, and the functional component in terms of illocutionary and interactive function. In fact, the grammar rules of the different models are not sufficient to direct and to control the dialog.

The presented dialog model combines the structural and functional components. The conversational (interactional, sequential and structural) constraints of the Geneva model need to be taken into account during the dialog act identification. This model includes different sub-dialogs aligned on three axes, the task, the dialog and the meta-dialog. With respect to the hierarchy (exchange, intervention and language act), these sub-dialogs separate the different dialog phases, therefore enabling a precise control of the dialog flow. The identification of the sub-dialogs and the dialog acts has been carried out on the basis of corpus analyses, with the aim to offer a functional model, that may directly be used for a specific application.

Notes

[1] a succession of actions necessary to obtain a result.

[2] The expression *dialog act* has been introduced by Bunt (1994).

[3] This expression is used, in particular, by Cohen & Levesque. We note that relations such as *the globe is round* and *I think that it is not round* or *well, I intend to open the gate so that it is closed,* taken seriously, account for mental configurations that are not rationally balanced.

This page intentionally left blank

Chapter 4

CONCLUSION

Speech recognition, understanding, dialog and speech synthesis capabilities render the interaction between humans and computers more efficient and natural. If the machine not only recognizes, but also understands the spoken natural language input, and this even for multiple languages, an easy access to a wide range of information and communication services is granted.

Spoken natural language engineering helps to achieve these aims. Some may already be realized, although they still require some improvements. Significant progress has been made in this field and the realizations should follow up in the next few years.

It seems obvious that understanding and dialog interaction are paramount stages, in particular for information retrieval and database access applications. Nevertheless, the expansion of these new communication modes cannot become effective unless reliability and usability become acceptable for the general public.

The presented methods to dialog interaction are part of the permanent research efforts carried out by the scientific community in human-machine communication. In fact, spoken language dialog modeling constitutes a scientific problem that is of interest to an increasing number of researchers, notably because of recent advances in speaker-independent recognition. The human-machine spoken language dialog problem is not a completely new field, if we consider the research carried out in linguistics and written language.

We have described the integration of a rule-based semantic analysis component into a human-machine spoken language dialog system that operates within an application for information requests. At the dialog management level, we have presented the aspects of task and dialog modeling.

One approach to task and dialog modeling consist of separating the task model, which is directly related to the application, from the dialog model, which rather describes the dialog characteristics for a given class of application). The presented task model applies a unified, flexible and powerful formalism of rules encapsulated in task and plan structures. However, both models are not entirely independent. The task model communicates certain information to the dialog model. If, for example, the system needs to obtain or confirm certain parameters required for a particular task execution, the task model generates a dialog act of request for precision or confirmation.

The task modeling enables completion of the understanding mechanism by contextually interpreting the user utterances, by correcting the semantic case grammar representations, if necessary, and by calculating default values. Based on a library of predefined plans, the task model transfers information to the dialog model indicating the questions to be asked for or the confirmations to be gathered from the user. The task model also generates commands towards the application back-end and indicates to the dialog model the type of user feedback to be formulated.

Due to its complexity the dialog modeling is tackled from various scientific viewpoints, i.e. from philosophy and computer science to sociology, psychology, linguistics, etc. The described approach to dialog modeling is based jointly on philosophical and linguistic theories, and may be implemented using formal methods in computer science. In fact, it appears difficult to directly use linguistic models, since they do not pursue the same objectives. But it seems very important to rely the research on complementing linguistic studies. The described approach primarily falls within the crossroads of linguistics and computer science. The dialog model clearly shows the distinction between task and dialog, which seems significant not only from a theoretical but also an engineering point of view.

The described method allows a refined dialog control to dynamically modify the dialog strategy and to start appropriate processes depending on the dialog structure. The presented approach to dialog modeling is thus in line with the structural models that result from the linguistic work of the Geneva University. The linguistic Geneva model is a hierarchical and functional dialog model aiming at the creation of a theoretical framework to dialog analyses.

Other work on human-machine spoken language dialog modeling uses either models that result from linguistic research, or models that issue from artificial intelligence, like planning or modal logic.

The described research offers a formal framework of dialog modeling where structure and function are merged. The model includes different sub-dialogs. In fact, it seems convenient in spoken language dialog management to identify the dialog acts and the different sub-dialogs, and to organize them hierarchically. This allows the machine a refined dialog control and a relevant feedback generation to the user. Furthermore, the initiative and reactive parts of the dialog are separated. For example, if as a reply to a system request for precision, an explanation request is uttered by the user, it seems appropriate that the system satisfies this request for explanation first, before eventually restarting the precision request.

Using dialog act identification, the model is able to indicate the exact dialog state. Such a process enables the system to identify the function that each user utterance holds in the context of the dialog. Dialog strategies, as well as the response generation performed by the system may be separated depending on the dialog state. For example, the type of formulation differs depending on whether a request for reformulation intervenes after a request for precision or a contestation. Similarly, according to the number of repetitions the system needs to ask for, the formulation may be more precise and insisting.

Through its refined structure the dialog model allows anticipation/prediction in the dialog, in terms of dialog acts. This prediction may be propagated down to the lower language levels, such as speech recognition and understanding, by associating the possible surface forms to each act. In the model, no explicit information on the task contents exists. The dialog model yields the advantage of being able to be used in different applications for information request.

Nevertheless, the dialog model makes the assumption of a standard and cooperative user, without taking his intentional aspects into account, since for now no integrated user modeling is applied.

The dialog act concept may be generalized to a communication act, in order to use such a modeling approach for multimodal dialogs, or other types of applications. Further dialog corpora analyses would enable the identification of all relevant features that may contribute to the dialog act recognition, such as the syntactic structure (words, turnings, etc.). The dialog model may be fully exploitable to adapt the dialog strategy, as well as the surface forms that depend on the dialog state.

This page intentionally left blank

References

Aho, A. and Ullman, J. (1972). *The Theory of Parsing, Translation and Compiling,* volume 1. Prentice-Hall Inc.

Allen, J. (1988). *Natural Language Understanding.* The Benjamin/Cummings Publishing Company, Inc.

Allen, J. and Perrault, C. (1980). Analysing intention in utterances. *Artificial Intelligence,* 15:143–178.

Austin, J. (1962). *How to do thing with words.* Clarendon Press, Oxford.

Bennacef, S., Devillers, L., Rosset, S., and Lamel, L. (1996). Dialog in the RailTel Telephone-Based System. In *Proceedings of International Conference of Speech and Language Processing, ICSLP,* pages 550–553.

Bennacef, S., Néel, F., and Bonneau-Maynard, H. (1995). An Oral Dialogue Model Based on Speech Acts Categorization, Workshop on Spoken Dialogue Systems. In *ESCA Workshop on Spoken Dialogue Systems*, pages 237–240.

Bennacef, S. K. (1995). *Modélisation du dialogue oral Homme-Machine - Mise en œuvre dans une application de demande d'informations.* PhD thesis, Université de Paris XI, Orsay.

Bennacef, S. K., Bonneau-Maynard, H., Gauvain, J. L., Lamel, L. F., and Minker, W. (1994). A Spoken Language System For Information Retrieval. In *Proceedings of International Conference of Speech and Language Processing, ICSLP,* pages 1271–1274.

Bilange, E. (1991). An approach to oral dialogue modelling. In *Second Venaco Workshop on the structure of multimodal dialogue,* pages 1–12.

Bilange, E. (1992). *Dialogue personne-machine, modélisation et réalisation informatique.* Hermès.

Bonneau-Maynard, H., Gauvain, J. L., Goodine, D., Lamel, L. F., Polifroni, J., and Seneff, S. (1993). A French Version of the MIT-ATIS System: Portability Issues. In *Proceedings of European Conference on Speech Technology, EUROSPEECH,* pages 2059–2062.

Bruce, B. (1975). Case Systems for Natural Language. *Artificial Intelligence,* 6:327–360.

Bunt, H. (1989). Information dialogues as communicative action in relation to information processing and partner modelling. In M.M. Taylor, F. N. and Bouwhuis, D., editors, *Structure of Multimodal Dialogue.* North Holland.

Bunt, H. (1994). Context and Dialogue Control. *Think Quartely,* 3(1):19–31.

Caelen, J. (1994). Modélisation de la tâche et de l'usager. Technical report, GDR-PRC. Rapport intermédiaire du projet DALI.

Chapanis, A. (1979). Interactive Communication: A few research answers for a technological explosion. In *Nouvelles Tendances de la Communication Homme-Machine,* pages 33–67. M.A. Amirchahy et D. Néel, Inria, Orsay.

Cohen, P. and Perrault, C. (1979). Elements of Plan-based Theory of Speech Acts. *Cognitive Science*, 3:177–212.

Deutsch, B. G. (1974). The structure of Task Oriented Dialogs. In *IEEE Symp. Speech Recognition, CMU.*

Ehrlich, U., Hanrieder, G., Hitzenberger, L., Heisterkamp, P., Mecklenburg, K., and Regel-Brietzmann, P. (1997). ACCeSS - Automated Call Center Through Speech Understanding System. In *Proceedings of European Conference on Speech Technology, EUROSPEECH,* pages 1823–1826.

Ermine, J. (1993). *Génie logiciel & génie cognitif pour les systèmes à base de connaissances.* Lavoisier.

Ferrari, S. (1994). Modèle de la Tâche dans le cadre d'une interface de dialogue multimodal. In *6èmes journées sur l'Ingénierie des Interfaces Hommes-Machines,* pages 49–54.

Ferrari, S. (1997). *Méthode et outils informatiques pour le traitement des métaphores dans les documents écrits.* PhD thesis, Université de Paris XI.

Fillmore, C. J. (1968). The case for case. In Bach, E. and Harms, R. T., editors, *Universals in Linguistic Theory,* pages 1–90. Holt and Rinehart and Winston Inc.

Gauvain, J. L., Bennacef, S., Devillers, L., Lamel, L., and Rosset, S. (1997). Spoken Language Component of the MASK Kiosk. In Varghese, K. and Pfleger, S., editors, *Human Comfort & Security of Information Systems,* pages 93–103. Springer-Verlag.

Gavignet, F., Guyomard, M., and Siroux, J. (1992a). Mise en œuvre d'une application multimodale orale et graphique : le projet GÉORAL. In *Séminaire dialogue du GDR-PRC CHM sur le dialogue.*

Gavignet, F., Guyomard, M., Siroux, J., and Bretier, P. (1992b). Le projet multimodal : GÉORAL. In *Séminaire dialogue du GDR-PRC CHM sur le dialogue.*

Grau, B. and Vilnat, A. (1997). Cooperation in Dialogue and Discourse Structure. In *IJCAI workshop on Collaboration, Cooperation and Conflict in Dialogue Systems.*

Greibach, S. (1965). New Normal-Form Theorem for Context-Free Phrase Structure Grammars. *Journal of the Association for Computing Machinery,* 12(1):42–52.

Grice, H. (1975). Logic and Conversation. In *Syntax and semantic - Vol. III : Speech Acts,* pages 41–58. Academic press.

Guyomard, M. and Siroux, J. (1988). Une approche de la coopération dans le dialogue oral homme-machine. In *Colloque ERGO-IA,* pages 287–301.

Guyomard, M., Siroux, J., and Cozannet, A. (1990). Le rôle du dialogue pour la reconnaissance de la parole. In *Proceedings of Journées d'Études en Parole, JEP,* pages 322–326.

Hayes, P., Hauptman, A., Carbonnell, J., and Tomita, M. (1986). Parsing Spoken Language, a Semantic Caseframe Approach. In *Proceedings of Conference on Computational Linguistics, COLING,* pages 587–592.

Issar, S. and Ward, W. (1993). CMU's Robust Spoken Language Understanding System. In *Proceedings of European Conference on Speech Technology, EUROSPEECH,* pages 2147–2150.

Lamel, L., Rosset, S., Gauvain, J., Bennacef, S., Garnier-Rizet, M., and Prouts, B. (1998). The LIMSI ARISE system. In *Interactive Voice Technology for Telecommunications Applications, IVTTA,* pages 209–214.

Lamel, L. F., Bennacef, S. K., Bonneau-Maynard, H., Rosset, S., and Gauvain, J. (1995). Recent Developments in Spoken Language Systems for Information Retrieval. In *Proceedings of ESCA Workshop on Spoken Dialogue Systems,* pages 17–20.

Litman, D. and Allen, J. (1987). A Plan Recognition Model for Subdialogues in conversations. *Cognitive Science,* 11:163–200.

Litman, D. and Allen, J. (1990). Discourse Processing and Commonsense Plan. In *Intentions in Communication,* pages 365–388. MIT Press.

Luzzati, D. (1995). *Le dialogue verbal homme-machine, études de cas.* Masson.

Marque, F., Bennacef, S., Néel, F., and Trinh, S. (1993). PAROLE : A Vocal Dialogue System For Air Traffic Control Training. In *Applications of Speech Technology.*

Matrouf, A., Gauvain, J., Néel, F., and Mariani, J. (1990a). Adapting Probability-Transitions in DP Matching Process for an Oral Task-Oriented Dialogue. In *Proceedings of International Conference on Acoustics Speech and Signal Processing, ICASSP.*

Matrouf, A., Gauvain, J., Néel, F., and Mariani, J. (1990b). An Oral Task-Oriented Dialogue for Air-Traffic Controller Training. In *SPIE-IEEE Applications of Artificial Intelligence.*

Matrouf, A. and Néel, F. (1991). Use of upper level knowledge to improve human-machine interaction. In *VENACO WORKSHOP AND ETRW on multimodal dialogue,* pages 1–14.

Minker, W. and Bennacef, S. (2000). *Parole et Dialogue Homme-Machine.* Collection Sciences et Techniques de l'Ingénieur. CNRS Editions & Éditions Eyrolles, Paris.

Minker, W., Waibel, A., and Mariani, J. (1999). *Stochastically-based Semantic Analysis,* volume 514. Kluwer Academic Publishers, The Kluwer International Series in Engineering and Computer Science.

Minsky, M. (1975). A Framework for Representing Knowledge. In Winston, P. H., editor, *The Psychology of Computer Vision,* pages 211–277. McGraw-Hill.

Moeschler, J. (1985). *Argumentation et conversation.* Hatier-Paris.

Moeschler, J. (1989). *Modélisation du dialogue. Représentation de l'inférence argumentative.* Hermès.

Moeschler, J. (1991). L'analyse pragmatique des conversations. *Cahiers de Linguistique Française,* (12):7–30.

Morell, A. M. (1988). *Analyse Linguistique d'un Corpus de Dialogues Homme-Machine,* volume I - Premier corpus: Centre de Renseignements SNCF à Paris. M.A. Amirchahy et D. Néel, Inria, Orsay.

Néel, F., Chollet, G., Lamel, L., Minker, W., and Constantinescu, A. (1996). Reconnaissance et Compréhension - Évaluation et Applications. In *Fondements et perspectives en Traitement Automatique de la Parole,* pages 331–367. AUPELF-URF.

Ostler, N. (1989). LOQUI : How Flexible can a Formal Prototype Be ? In *The Structure of Multimodal Dialogue,* pages 407–416. Elsevier Science.

Peckham, J. (1993). A New Generation of Spoken Dialogue Systems: Results and Lessons from the Sundial Project. In *Proceedings of European Conference on Speech Technology, EUROSPEECH,* pages 33-40.

Pernel, D. (1991). Le corpus 'PLUS'. Technical report, LIMSI-CNRS. Notes et documents 91-19.

Pernel, D. (1994). *Gestion des buts multiples de l'utilisateur dans un dialogue Homme-Machine d'informations.* PhD thesis, Université de Paris XI.

Pouteau, X., Krahmer, E., and Landsbergen, J. (1997). Robust Spoken Dialogue Management for Driver Information Systems. pages 2207–2211.

Price, P. (1990). Evaluation of Spoken Language Systems: The ATIS Domain. In *Proceedings of DARPA Speech and Natural Language Workshop,* pages 91–95.

Proctor, C. and Young, S. (1989). Dialogue Control in Conversational Speech Interfaces. In *The structure of multimodal dialogue,* pages 385–398. North Holland.

Rabiner, L. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of IEEE,* 77(2):257–285.

Roulet, E. (1981). Échanges, interventions et actes de langage dans la structure de la conversation. *Études de linguistique appliquée,* (44):7–39.

Roussanaly, A. and Pierrel, J. (1992). Dialogue oral Homme-machine en langage naturel : le projet DIAL. *TSI,* 11(2):45–91.

Sabah, G. and Briffault, X. (1993). CARAMEL : a Step towards Reflexion in Natural Language Understanding systems. In *IEEE International Conference on Tools with Artificial Intelligence,* pages 258–265.

Sadek, D. and Mori, R. D. (1997). Dialogue Systems. In *Spoken Dialogues with Computers, R. De Mori (ed.),* pages 523–561. Academic Press.

Searle, J. (1969). *Speech Acts.* Cambridge University Press.

Searle, J. (1979). *Expression and Meaning.* Cambridge University Press.

Searle, J. (1983). *Intentionality.* Cambridge University Press.

Seneff, S. (1992). TINA: A Natural Language System for Spoken Language Applications. *Computer Speech and Language,* 18(1):61–86.

Seneff, S., Hurley, E., Lau, R., Paoa, C., Schmid, P., and Zue, V. (1998). Galaxy-II: A Reference Architecture for Conversational System Development. In *Proceedings of International Conference of Speech and Language Processing, ICSLP.*

Siroux, J., Guyomard, M., Jolly, Y., Multon, F., and Remondeau, C. (1995). Speech and tactile-based GEORAL system. In *Proceedings of European Conference on Speech Technology, EUROSPEECH.*

Vanderveken, D. (1988). *Les actes de discours.* Pierre Mardaga éditeur.

Vilnat, A. and Nicaud, L. (1992). Un système de dialogue Homme-Machine: Standia. In *Séminaire du GDR-PRC CHM sur le dialogue,* pages 85–99.

Wachtel, T. (1986). Pragmatic sensitivity in NL interfaces and the structure of conversation. In *Proceedings of Conference on Computational Linguistics, COLING,* pages 35–41.

Walker, M., Hirschmann, L., and Aberdeen, J. (1999). Evaluation for DARPA COMMUNICATOR Spoken Dialogue Systems. In *LREC Workshop on Multimodal Resources and Multimodal Systems Evaluation.*

Ward, W. (1994). Extracting Information in Spontaneous Speech. In *Proceedings of International Conference of Speech and Language Processing, ICSLP,* pages 83–86.

Waterworth, J. (1982). Man-Machine 'speech dialogue acts'. In *Applied Ergonomics.*

Wilensky, R. (1981). Meta-Planning : Representing and Using Knowledge About Planning in Problem Solving and Natural Language Understanding. *Cognitive Science,* 5:197–233.

Wilensky, R. (1983). *Planning and Undestanding.* Addison Wesley.

Yamoka, T. and Iida, H. (1990). A Method to Predict the Next Utterance using a Four-layered Plan Recognition Model. In *European Conference on Artificial Intelligence, ECAI,* pages 726–731.

Young, S., Hauptmann, G., Smith, E., and Werner, P. (1989). High Level Knowledge Sources in Usable Speech Recognition Systems. In *Communications of the ACM,* pages 183–194.

Young, S. and Proctor, C. (1989). The Design and Implementation of Dialogue Controle in Voice Operated Database Inquiry Systems. *Computer Speech and Language,* 3:329–353.

Young, S. and Ward, W. (1988). Towards Habitable Systems : Use of World Knowledge to Dynamically Constrain Speech Recognition. In *Second Symposium on Advanced Man-Machine Interface.*

Young, S. J. (1992). *HTK V1.4 User, Reference Programmer Manual.* Cambridge University Engineering Dept, Speech Group.

About the Authors

Wolfgang Minker is a full-time Professor at the University of Ulm, Department of Information Technology. He received his Ph.D. in Engineering Science from the University of Karlsruhe (Germany) in 1997 and his Ph.D. in Computer Science from the University of Paris-Sud (France) in 1998. He has been Researcher at the Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur (LIMSI-CNRS), France, from 1993 to 1999 and member of the scientific staff at DaimlerChrysler, Research and Technology (Germany) from 2000 to 2002.

Samir Bennacef is Research Engineer at Vecsys (France) where he is working on human-machine dialogue and the design of vocal servers for information retrieval systems. He received his Ph.D. in Computer Science from the University of Paris-Sud (France) in 1995. He has been Researcher at the Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur (LIMSI-CNRS), France, from 1991 to 1998.

This page intentionally left blank

Index

act, 3, 30, 31, 33, 35, 43–46, 60, 62, 67, 72–75, 77, 83
- closing formality, 65
- communication, 44, 83
- confirmation request, 66
- contestation, 64
- correction, 65
- dialog, 28, 29, 37, 38, 40–43, 46, 47, 59, 60, 67–69, 72–75, 77–79, 82, 83
- discussion, 64
- explanation request, 62
- illocutionary, 30–32
- information request, 61, 62, 64
- interrogative, 40
- language, 2, 23, 29–37, 39, 40, 43–46, 51, 59, 72, 78, 79
- locutionary, 30
- opening formality, 61
- perlocutionary, 30
- precision request, 62
- reformulation request, 66
- reply, 64, 66
- request, 39
- response, 39, 62, 64, 67, 72
- resumption, 62
- speech, 30
- standby, 63

analysis, 8, 13, 19, 20, 32–34, 43, 48, 51
- case frame, 15, 16
- corpus, 51, 7 2, 79, 83
- deep, 42
- dialog, 43, 46, 52, 60, 68, 70, 72, 82
- discourse, 33, 34
- functional, 34
- hierarchical, 34
- language, 77
- linguistic, 60
- manual, 21
- morphological, 74
- pragmatic, 46
- semantic, 10, 81
- spoken language, 35
- surface, 42
- syntactic, 14, 21
- syntactic-semantic, 46, 74, 77
- utterance, 33, 49, 77

case, 14–17, 19, 20
- frame, 14–17, 19–21
- grammar, 2, 14–17, 27, 28, 60, 72, 74, 82
- marker, 15, 17, 19
- system, 15

communication, 1–6, 9, 29, 39, 44, 46, 49, 57, 61, 74
- act, *see* act, communication
- action, 44
- channel, 5–7, 12, 57, 68
- flexible, 6
- human-machine, 2, 4, 5, 47, 81
- management, 38
- mode, 4, 5, 74, 81
- multimodal, 5
- quality, 57
- service, 81
- sign, 3

communicative function, 32, 46

concept, 2, 6, 8, 14–17, 23, 24, 26, 27, 30, 31, 33, 42–44, 47–49, 60, 72, 77, 78, 83

constative utterance, 29

constraints
- conversational, 34, 43, 79
- interactional, 34
- linguistic, 6
- linkage, 34

structural, 34, 41, 72
syntactic, 6, 17, 48
contextual interpretation, 6, 7, 12, 28, 49, 50
conversation, 9, 10, 31, 33–38, 41, 43, 51, 52, 72

database, 2, 10, 19, 48, 52, 63
access, 1, 4, 10, 20, 21, 46, 48, 52, 57, 61, 81
interface, 10, 16
query, 10, 16, 20
retrieval, 67
dialog, 1–13, 16, 24, 28, 29, 32, 34–53, 55–63, 65–68, 70, 72–79, 81–83
act, *see* act,dialog
control, 6, 48, 59, 72, 73, 82
corpus, 2, 23, 29, 51, 52, 83
flow, 2, 5, 9, 23, 25, 34, 37, 39, 46, 50, 59, 72, 78, 79
grammar, 37, 68
history, 9, 10, 16, 49, 74, 76, 78, 79
level, 39
management, 8, 10, 12, 23, 49, 52, 59, 73, 74, 81, 82
meta-, *see* meta-, dialog
model, *see* model, dialog
plan, *see* plan, dialog
sub-, *see* sub-dialog
system, 4, 7, 9–11, 24, 25, 34, 44, 47–50, 60, 73–75, 81

exchange, 34, 36, 38–43, 49, 51, 60, 72, 79
sub-, 36, 38

formality, 42, 53, 55, 57
act, *see* act, formality
closing, 56, 60, 65,69
negotiation, 42
opening, 55, 60, 61, 65, 69, 75, 77
politeness, 42
sub-dialog, *see* sub-dialog, formality
functional approach, 26

grammar, 30, 35, 37, 41, 43, 48, 51, 60, 68
axiom, 68
case, *see* case, grammar
context-free, 21, 48
dialog, *see* dialog, grammar
formal, 35, 60
regular, 68
rules, 68, 72, 79
semantic, 14, 21
slot, 15
theory, 51

illocutionary, 34, 72
act, *see* act, illocutionary
aspect, 32
force, 30–32
function, 34, 40, 60, 72, 79
logic, 31
objective, 31
response, 40
inference, 8, 14, 47
information evaluation, 46
initiative, 39–41, 43, 56, 59, 67, 73, 82
movement, *see* movement, initiative
intervention, 24, 34, 36, 40–43, 50, 51, 57, 58, 72, 79
composed, 40
simple, 40

language, 2, 3, 6, 8, 9, 11–13, 29, 31, 32, 37, 44, 81
-dependent, 77
-independent, 77
act, *see* act, language
analysis, *see* analysis, language
coverage, 6
database access, 4
declarative, 17
effects, 21
engineering, 81
formal, 13, 51
level, 7, 73, 83
meaning, 32
model, 8
natural, 1, 2, 4, 5, 11, 12, 81
ordinary, 29
philosophy, 31
production of, 47
programming, 4, 13
response, 16, 21, 23
spoken, 1–8, 10, 11, 14, 16, 23, 24, 28–31, 35, 39, 40, 42–44, 47, 48, 50, 52, 55, 59, 60, 66, 67, 73–75, 79, 81, 82
structure, 32
studies, 3
style of, 77
system query, 16
theory, 2, 23, 68
understanding, 12, 13, 15, 16, 18, 52
written, 5, 81
level, 8, 9, 14, 15, 17, 18, 26, 27, 34, 43, 49, 52, 63, 78, 81
basic, 18, 19
cognitive, 7, 50
common, 17, 18
conceptual, 17, 18
dialog, *see* dialog, level
formal, 46
formality, 42
incidential, 39
intermediate, 17, 18

intervention, 42
language, *see* language, level
lexical, 34, 77
main, 39, 53
meta-, 58
phrase, 33
recursivity, 69
reference, 46
secondray, 39
semantic, 77
semantic-pragmatic, 49
socio-cultural, 7
structural, 42
syntax, 33
user expertise, 7
vocabulary, 6

locutionary act, *see* act, locutionary

marker, 14, 15, 17, 19, 20
case, *see* case, marker
illocutionary force, 32
linguistic surface, 79
non-, 18, 19
non-adjacent, 18, 19
post-, 17, 19
pre-, 17, 19
semantic, 19

meta-
dialog, 52, 57, 67, 68, 72, 74, 75, 77–79
plan, 43
reasoning, 6, 7

mixed approach, 26

model, 1, 2, 8, 9, 23–26, 28, 33–36, 39, 40, 42–48, 51, 53, 59, 60, 63, 66, 68, 69, 72, 73, 76, 79, 82, 83
cognitive, 13
dialog, 2, 9, 11, 23, 28, 29, 34–41, 43, 47, 49–51, 59, 60, 66–70, 72–74, 78, 79, 81–83
Geneva, 33, 34, 39, 40, 43, 60, 72, 79, 82
language, *see* language, model
logic, 29, 35, 45, 47
plan-oriented, 29, 35, 43, 44
structural, 2, 29, 35, 51, 79, 82
system, 9
task, 2, 8, 9, 23–29, 35, 48–50, 75–77, 81, 82
user, 9, 45, 46, 48, 49, 83

movement, 3, 36, 38, 46
initiative, 38
reactive, 38, 39
sub-, 38

non-grammatical, 5

over-information, 73

performative utterance, 29, 30

perlocutionary act, *see* act, perlocutionary

plan, 2, 6, 9, 25, 28, 35, 42–48, 62, 75, 78, 82
-oriented, 23, 43
communication, 44
dialog, 44
discourse, 44
domain, 44
execution, 78
hierarchy, 28
interaction, 44
meta-, *see* meta-, plan
model, *see* model, plan
multi-, 45
recognition, *see* recognition, plan
structure, 2, 23, 28, 82
task, 76
user, 44, 45

prediction, 8, 48, 49, 60, 73, 83

propositional content, 30, 31, 34, 35, 44, 60, 72, 79

reactive, 39, 40, 43, 59, 72, 73, 82
movement, *see* movement, reactive

recognition, 5, 8, 11, 48, 49, 59, 81
activity, 9
dialog act, 42, 79, 83
plan, 35, 44
speech, 5–12, 48, 49, 73, 74, 81, 83

structural approach, 26

sub-dialog, 36, 51–69, 72, 73, 79, 82
argumentation, 73
clarification, 5, 38, 39, 44
classification, 52
closing, 57
confirmation, 58, 66
contestation, 54, 61, 64
correction, 58, 65, 67
disambiguation, 58, 59, 65–67
discussion, 54, 61, 64
evaluation, 38, 39
explanation, 53, 62, 68, 73
formality, 55, 60, 65
information, 60–62
information request, 61, 62, 66
information-seeking, 53
labeling of, 59
management, 79
precision, 53, 54, 61, 62, 68
reformulation, 5, 66
repetion, 58
response, 53, 64
restart, 62
resume, 57
standby, 57, 61, 63, 67

transaction, 40, 41

WOZ, 16, 51, 52, 55