Multi-modal Interaction Management for a Robot Companion

Shuyin Li

M.A. Informationsverarb. Shuyin Li
AG Angewandte Informatik
Technische Fakultät
Universität Bielefeld
email: shuyinli@techfak.uni-bielefeld.de

Abdruck der genehmigten Dissertation zur Erlangung
des akademischen Grades Doktor-Ingenieur (Dr.-Ing.)
Der Technischen Fakultät der Universität Bielefeld
am 23.05.2007 vorgelegt von Shuyin Li
am 06.09.2007 verteidigt und genehmigt.

Gutachter:

Dr. Britta Wrede, Universität Bielefeld
Prof. Dr. David Traum, University of Southern California

Prüfungsausschuss:

Prof. Dr. Ipke Wachsmuth, Universität Bielefeld
Prof. Dr. Franz Kummert, Universität Bielefeld
Prof. Dr. Holger Theisel, Universität Bielefeld
Dr. Britta Wrede, Universität Bielefeld
Prof. Dr. David Traum, University of Southern California

Gedruckt auf alterungsbeständigem Papier nach ISO 9706

Multi-modal Interaction Management for a Robot Companion

Der Technischen Fakultät der Universität Bielefeld

zur Erlangung des Grades

Doktor-Ingenieur

vorgelegt von

Shuyin Li

Bielefeld – Mai 2007

Acknowledgments

This thesis would not have been possible without the support of many people. I want to first thank my advisor Dr. Britta Wrede for many inspiring discussions in the last 3.5 years that helped me to substantially improve my work. I'm also very grateful to Prof. David Traum who carefully reviewed this thesis and offered many constructive suggestions. Many thanks are due to Prof. Gerhard Sagerer who gave me the chance to finish my PhD in the group of Applied Computer Science at the Bielefeld University. Here I spent wonderful time with my colleagues who are all kind and cooperative teammates, especially those in the BIRON-group.

On the personal side, I would like to thank my parents who encouraged me to pursue my study and research in Germany although this means that they have to be 9000 km away from their youngest daughter. I also wish to thank Marcus Kleinehagenbrock who supports me both emotionally and professionally and gives me a home in Germany.

And I also wish to thank our robot BIRON for which I developed the Interaction Management System. I know how hard BIRON is trying to acquire human abilities. By working at BIRON, I have learned to appreciate human beings and all their sophisticated capabilities. I also wish to express my deep respect for science and the tireless endeavor of human beings to learn about themselves and the environment they live in.

Contents

1. Introduction

What is your dream robot?

Maybe you would think of a robot specialized in cooking spaghetti, or maybe a robot with the face of Cameron Diaz, or you even want to have an emotional robot that behaves the same way as your lost dog. Actually, many of these issues are subjects of scientific as well as non-scientific debates, which are further heated by contributions of the film industry in form of various, imaginative “what-if-scenarios”. The central issue of these debates is essentially the one single question: “Should a robot be like a human?” While the discussions concerning robot appearance, emotional abilities etc sometimes become intense, few people put into question whether a robot should have human-like interaction capabilities. The author can not remember that a (fictive) robot become a “film star” although its film partners have to interact with it by pressing buttons. This is probably because that interaction capabilities are so fundamental for humans as social being that we take it for granted for any intelligent systems. However, exactly for the same reason, it is highly challenging to actually realize these capabilities so that they comply with our own standard as masters of communication. The current work attempts to go a step forward towards the realization of sophisticated interaction capabilities for robots that are meant to accompany us in our everyday life.

1.1. Human-Robot Interaction (HRI)

Human-robot interaction is the subject that concerns the study of interaction between human users and robots. Although the robotics research started already more than 40 years ago, HRI is still an emerging field. The reason for the delayed start of HRI research can be found in the history of robotics research. In this section, a short history of robotics is first presented in section 1.1.1 and the main characteristics of HRI are then discussed in section 1.1.2.

1.1.1. A short history of robotics

A mechanical servant that can work autonomously and intelligently has been the wish of human beings since the ancient Greece. The realization of this idea, however, was not possible before the industrialization and scientific progress made in computer science.

The first robots that were actually developed were *industrial robots* in the 1960s, mostly robot arms, that were meant to complete repetitive or dangerous tasks in factories. Later, the robotics benefited from artificial intelligence and robots that can reason about their own actions and environments emerged. *Professional service robots* for fields like medical service or rescue operations have been developed to work with professional users. In the 1990s, the appearance and mobility of robots were greatly improved so that it was at least theoretically possible for them to co-exist with human beings. Since then *personal robots* have been emerging that are designed to serve or entertain humans in their everyday life. Figure 1.1 illustrates the evolution of robot applications since the 1960s and Fig. 1.2 presents several milestone robots.

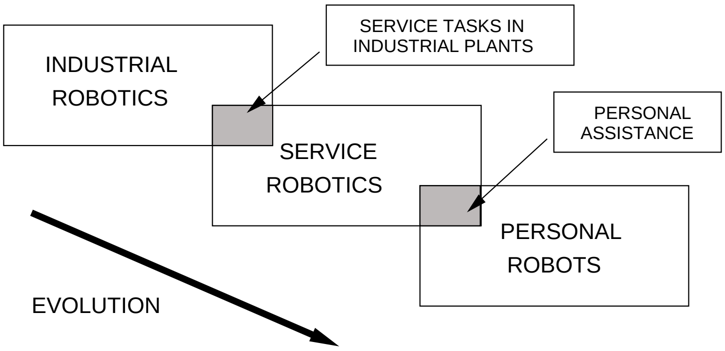


Figure 1.1.: A scheme for the evolution of robotics since 1960s, outlining three main areas as three basic steps of evolution [GLD99]

The evolution of robot applications has moved the main field of the research from well-controlled industrial environments to dynamic, real-life environments and from robot operation by well-trained professional users to robot interaction with untrained non-professional users. HRI research has begun to grow only with personal robots and concerns the robot interaction with non-professional users in dynamic, real-life environments.

Personal robots developed today fulfill various functions. Entertainment robots such as Aibo [Aib] and QRIO [Qri] are rather intelligent toys. Tour-guide robots, receptionist robots, hospital robots, e.g.,MINERVA [TBB+99], GRACE [SBG+03] and Hygeiorobot [SAS01], perform pre-defined tasks in pre-defined environments and often only have limited interaction capabilities. What these robots have in common is that they do not serve a specific human user and the interaction with such a robot is often on a short-term basis. This is different from the so-called *mobile robot companions*: They are intended to serve human users in their household on a long-term basis and should be able to *perform useful tasks, acquire new knowledge and behave socially* [DWK+05]. The realization of such an autonomous robot is highly challenging and requires the combination of a number of advanced technologies: robust perception of physical and social environments, sophisticated reasoning about users' and robot's own activities, human-like interaction capabilites, and so on.

The current work concerns the development of an interaction management system that enables human-like interaction for a robot companion. To achieve this goal, it is important to know

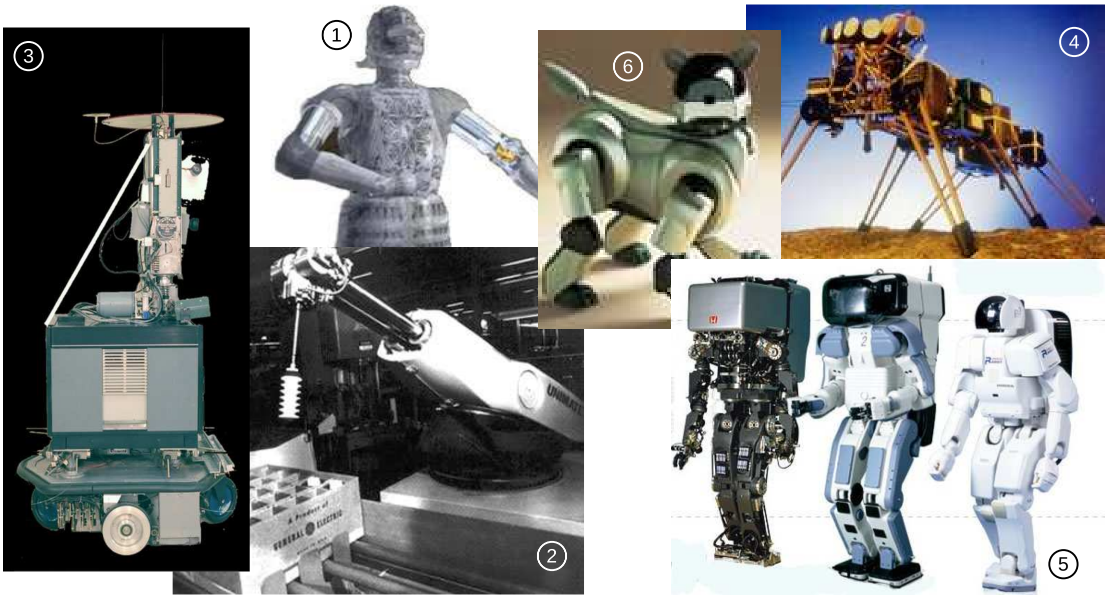


Figure 1.2.: Milestone robots. ① Robot designed and possibly built by Leonardo da Vinci in approximately 1495, it is an outgrowth of his earliest anatomy and kinesiology studies ② Unimate (1961), one of the first industrial robot arm. It is controlled step-by-step by commands stored on a magnetic drum. ③ Shakey (1966 - 1972), the first mobile robot that is able to reason about its actions and environments [Nil84]. ④ Genghis (1989), one of the first walking robots [Bro89]. ⑤ P1, P2 and P3 (1993 -1997), humanoid robots developed by HONDA. ⑥ Aibo (1999), a robotic pet by SONY.

the characteristics of interaction with such a robot and their impact on interaction design. This point is discussed in the next section. Below, whenever HRI is mentioned, it refers to interaction between a human user and a robot companion.

1.1.2. Characteristics of HRI

The interaction management system of a robot companion has the responsibility to communicate its decisions and actions with its human user in a natural way. When putting these responsibilities in concrete terms, special characteristics of such a robot must be taken into account. Below, two characteristics of a robot companion, which they inherit from mobile robots in general, are first discussed: situatedness and embodiment. Then, the characteristics of tasks and potential users of a robot companion are presented. These characteristics are rather unique for robot companions and distinguish them from other types of robots.

Situatedness: Most desktop and virtual agent applications (see section 2.2.2) represent a virtual world that is relatively predictable. For example, the existence of obstacles, variation of lighting conditions, and so on can be easily controlled. Additionally, concepts like time, space, weight, distance etcetera are abstract and there is usually no serious consequence when laws of physics are violated. In contrast, a robot is situated “here and now” [Bro86, Bro89] and cohabits the same physical, real world as a human. This means, firstly, a robot has to deal with less predictable environments. In case of the interaction, even the recognition of whether the user has initiated an interaction becomes a challenging issue: it can be confused with radio, TV, conversations between two persons sitting nearby and so on. Thus, the interaction management system of a robot should be able to facilitate the interaction recognition process. Further, in the real environment, laws of physics govern activities of a robot as they govern those of a human. For example, there will be serious consequence for a robot and/or its environments if it tries to carry an object of twice the weight as itself, or moves to another room not by going through the door but by hitting itself against the wall. The situatedness requires a robot to be aware of its physical environments and to acquire human’s ability to deal with physical restrictions. One of the humans’ strategies to handle physical restrictions is to compensate their own inabilities with the abilities of others by asking for help. The employment of this strategy requires the interaction management system of a robot to realize interaction in a mixed-initiative style.

Embodiment: The embodiment [DJ00] of a robot companion changes the needs and the way of interaction. Empirical studies [NRSC03] show that the visual access to the body of one’s interaction partner affects the interaction in the way that non-verbal behaviors are used as communicative signals. For example, to refer to a cup that is visible to both a user and her robot, the user tends to say “this cup” and points to it. For the interaction management system of a robot, this means that it should account for multi-modal interaction. Further, multi-modality is also beneficial in situations in which certain modalities are temporarily unavailable or one modality is less effective than the other. For example, if a robot moves away, its display, a popular modality in general Human-Computer Interaction (HCI) applications, is no longer visible to its user and she may need to use speech modality instead. Speech is generally welcome in HRI [Kha98, LKF+04], however, it can become cumbersome when being used to describe spatial information [CG04].

Characteristics of tasks: A robot companion is intended to, among others, provide service for users to ease their burden of household work. These tasks can often be equally performed by some family members themselves, they probably even know better how to do them. This characteristic has the implication that a robot companion should be able to acquire new knowledge and skills through interaction with users. The learning ability is indispensable because each household is individual and it is hardly possible to specify all the knowledge and skills that a robot companion needs before head. This means that the interaction should be modeled relatively independently of the domain knowledge. Further, learning affects the relationship between a user and her robot: Learning through interaction is a cooperative process because both the teacher and the student work towards the same goal. This means, the relationship between a user and her robot should be viewed as cooperative partners rather than in a master-slave style.

Characteristics of users: Potential users of a robot companion usually do not have interaction experience with it initially and their view of robots are strongly influenced by science fiction films. Studies [Kha98] show that many of potential users have doubt as to whether they want to to have a robot as a “companion”. The reason is often insufficient trust in robot’s social abilities and flexibility. This concern is justified especially when users should live with such a robot on a long-term basis. For example, a robot that permanently interrupts a user’s conversation with other persons is hardly acceptable. To increase the acceptability of a robot companion, it should be able to demonstrate social awareness and act accordingly. The interaction management system is the direct interface between a user and a robot and should, therefore, contribute to the realization of this ability by actively observing user behaviors and adapting its own. Besides, as direct interface to users, the interaction management system must also take care of usability, one of the most essential requirements for technical systems in general. Given that a robot is often a highly complex system with many subsystems, the issue as to how to communicate a robot’s internal states with untrained users in a easily understandable way is also a challenge for the interaction management system.

The four characteristics of HRI and the resulting requirements for an interaction management system for a robot companion are summarized in Table 1.1.

Characteristics of HRI	Required abilities of an interaction management system
situatedness	recognition of interaction initiated by users mixed-initiative interaction style
embodiment	handling multi-modality making use of different modalities in a meaningful way
learning through interaction required	handling interaction relatively independently of domain knowledge (separation of interaction from domain task execution) handling cooperative interaction
untrained, naive users	exhibiting social behaviors contributing to the usability of the robot system

Table 1.1.: Characteristics of HRI and the required abilities of an interaction management system for a robot companion

As shown above, HRI for robot companions poses a number of scientific questions that need to be addressed by its interaction management system. Such a system should generally account for human-like interaction so that untrained human users can easily communicate with the robot. In the next section, the general processing context and requirements of an interaction management system is discussed.

1.2. Interaction management systems for human-like interaction

An interaction management system (IMS) is usually a part of an *interaction system*, which controls the interface of a technical system to its users. In HCI, the most popular kind of user interface is probably the Graphical User Interface (GUI). It became the standard user interface for most HCI applications in the last decades. However, interaction systems with GUI do not really support human-like interaction. Given the characteristics of HRI, other concepts have to be found for interaction systems in this domain.

A major feature of human-human interaction is the usage of language. The ability of performing sophisticated dialog distinguishes human being from most other species on earth. This means that the interaction system of a robot companion should be speech-enabled. A speech-enabled interaction system is called a “dialog system”. Although dialog is often associated with speech, it is insufficient to only realize a spoken dialog system for a robot companion. As discussed in the previous section, non-verbal behaviors of dialog participants are also used as communicative signals in face-to-face interaction and they need to be taken into account, too. An interaction system that is speech-enabled and can handle multi-modal information is called “multi-modal dialog system”. Such a system is probably the most promising candidate as the interaction system for a robot companion. In the following, the general processing concept of a multi-modal dialog system is discussed.

In order to carry out dialog, a system needs to first recognize input signals from the user and understand their semantic meanings. These tasks are traditionally done by two different subsystems: *speech recognizer* and *speech understander*. The semantic representation of the input signals are forwarded to the *dialog management system* (DMS), which performs dialog planning, i.e., making decisions as to what to do and/or to say in the next step. This decision can result in sending commands to back-end applications or constructing a “concept” serving as the basis for speech output, or both. After a more or less elaborate output planning process, the concept is then translated to some text and synthesized into speech signals by a *speech synthesizer*. These signals are the feedback of the dialog system to user’s speech input. Of course this general processing concept does not only concern speech, but also other modalities. For example, instead of recognizing speech input from a user, a system can recognize her gestures and infer her intention from them. Similarly, a system can generate imitated facial expressions instead of speech output as feedback to a user. The general processing concept of a multi-modal dialog system in illustrated in Fig. 1.3.

In the above concept, the DMS is the heart of the entire dialog system because it determines the flexibility and efficiency of the dialog system to a great extent. Essentially, the IMS of a robot companion is comparable to a DMS, however, the responsibilities of an IMS go beyond those of a DMS. As summarized in Table 1.1, beside classical dialog-related responsibilities such as initiative regulation, an IMS is also in charge of generation of social behaviors, which would give the robot a personality, and realization of usability, which is traditionally a question of interface design instead of dialog modeling. An IMS requires a powerful and flexible dialog model as the basis and should be developed with users in the loop. The current work addresses both issues and its contribution is presented in the next section.

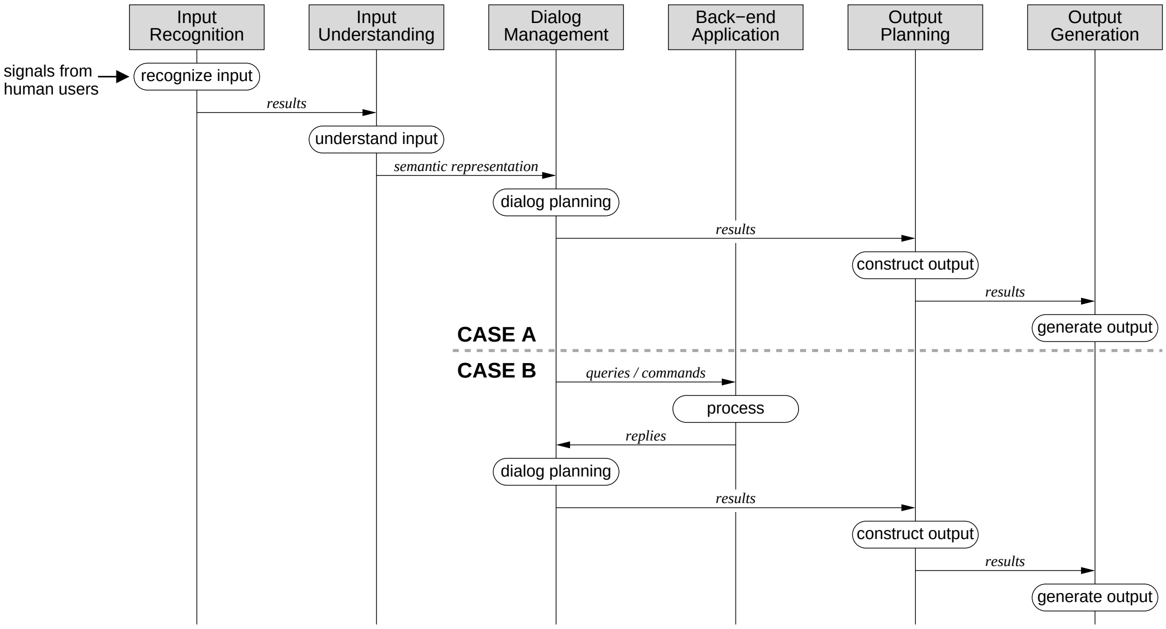


Figure 1.3.: General processing concept of a multi-modal dialog system as a UML sequence diagram: Case A = dialog processing without the involvement of the back-end application, Case B = dialog processing with the involvement of the back-end application

1.3. Contribution and outline of the current work

The goal of the current work is *to develop an interaction management system for a robot companion*. The major contribution of this work is twofold:

- A novel multi-modal dialog model is proposed, and
- Interactive behaviors were implemented following the pattern of Implementation-Evaluation-Cycle.

The new dialog model proposed in this work is a computational model of multi-modal grounding. Grounding is a well-known concept in dialog modeling research and refers to the process of establishing mutual understanding during an interaction. The new grounding model possesses two novel aspects: Firstly, this model improves existing grounding models and avoids many of their problems. Secondly, the grounding scheme is extended with the ability to directly handle multi-modality as well as pre-interaction contributions. This model is thus able to cover more dimensions of face-to-face interaction than many existing dialog models.

The multi-modal dialog model was implemented for the Interaction Management System of the prototype of robot companion BIRON. Given that the current work is one of the first attempts to develop sophisticated, multi-modal, interactive behaviors in HRI (see section 2.4), there were many open questions as to what behaviors should be implemented and how to evaluate them. Under this circumstance, implementation should not be viewed as the final step of the development process, but a part of an “Implementation-Evaluation-Cycle”. In the current work, observations were made during a user study that was conducted after the first version of the system had been implemented. These observations motivated the author to extend the implementation and to modify the experimental setup. The second version of the system was then evaluated in a second user study. In these two Implementation-Evaluation-Cycles, valuable insights into various aspects of HRI were gained, which greatly helped to establish comprehensive understanding of interaction modeling for robot companions. The employment of this iterative development concept also provided strong evidence for the powerfulness of the adopted dialog model because new interactive behaviors that had not been planned at the beginning could also be easily implemented without any modifications of the dialog model.

The current work is organized as follows:

Chapter 2 discusses existing works on multi-modal dialog modeling and its evaluation. The conclusion of this chapter is that the grounding concept is a promising candidate for the current purpose because it models face-to-face interaction by addressing laws governing dialog in general and it is sufficiently flexible.

Chapter 3 provides a detailed account of the novel dialog model. The discussions include comparison between existing works on grounding, key notions as well as operation rules of the new model and its evaluation.

Chapter 4 presents the Interaction Management System (IMS) of the robot BIRON that implements the new grounding model. In this chapter, details about the implementation platform, scenario, technical realization of the model are provided.

Chapter 5 presents how various interactive functions and behaviors are developed in the IMS through two Implementation-Evaluation-Cycles. In each of the cycles, the implemented behaviors are evaluated with a user study and the focus of the second cycle is built on the findings of the first one.

Chapter 6 summarizes the current work and discusses potentially beneficial extensions of the model and the system.

2. Foundational Work

This chapter presents the foundational work on multi-modal interaction management and its evaluation. Since the issues of sophisticated dialog modeling and multi-modality management have been the focus of two different research traditions, this chapter addresses them in two different sections: dialog modeling in section 2.1 and multi-modality management in section 2.2. Evaluation techniques in these two research directions are discussed in section 2.3. The research on interaction systems for HRI started much later than for Human Computer Interaction (HCI) applications so that there is a different picture of standard interaction systems in this field. Section 2.4 addresses this issue and presents state-of-the-art multi-modal dialog management strategies adopted in HRI.

2.1. Dialog modeling

The development of dialog modeling approaches has been the joint effort of speech technology and artificial intelligence since more than 30 years. Many approaches have been proposed in this period. McTear [McT04, McT02] categorizes these approaches into three classes: the finite state-based, the frame-based and the agent-based approach. This section provides a brief overview of the essential characteristics of the three main dialog modeling approaches today roughly based on McTear's categorization.

2.1.1. The finite state-based approach

The finite state-based approach is one of the first dialog modeling approaches. The basic idea is that a dialog can be represented as a state transition network in which states are determined by domain tasks. In each state the system carries out certain task-relevant action that is pre-defined. A typical action is asking the user a specific question to collect domain-relevant information. During a dialog the system can only be in one of these pre-defined task states. The transitions between these states determine all possible paths through the network, this means, the actions have to be done in certain pre-defined sequences. The user is expected to answer the system's question in each state to enable a legal transition. This way, the system arrives at its goal state, in which it performs the desired task, e.g., sending the user query to a database. Figure 2.1

illustrates such a transition network for the payment of a bill.

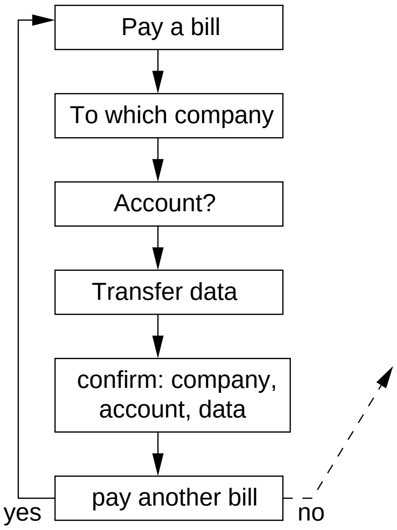


Figure 2.1.: An example dialog flow for payment of a bill ([McT02])

The biggest advantage of the finite state-based approach is its simplicity. It is particularly suitable for well-structured, relatively simple tasks, e.g., flight booking systems, train schedule information systems, book club services [AO95, LP99, LB94] and so on. Since the dialog management system has permanent control of the dialog flow the user's response is restricted. This point implies that the speech input of the user is more or less predictable and, thus, the performance of the speech recognition and speech understanding does not need to be of a very high standard. It is often sufficient to do key word spotting instead of complex continuous speech recognition. This advantage is well documented through comparative evaluation in simple task domains [PRBO96, DG95].

The finite state-based approach has two major disadvantages. From a functional view, this approach is not suitable for domains where the tasks are complex and not well-structured or the information needed for task execution has complex dependencies. Modeling complex tasks using finite state-based approach would mean that as many states as possible subtasks of the domain have to be pre-defined which can result in an unmanageable amount of states. Dependencies between parameters of the task can also lead to a “state explosion” as documented in [DBD98]. Besides, from the view of HCI, the finite state-based approach is very restricted and allows little freedom to the user. This problem becomes severe when users want to correct their preceding utterances or introduce some extra information.

2.1.2. The frame-based approach

The frame-based approach is similar to the finite state-based approach in the way that a predefined set of information needs to be collected for the task execution. The difference is, however, that this information collecting process does not need to happen in a pre-determined sequence which enables a greater flexibility in terms of mixed-initiative dialog style. A frame can be a simple data structure consisting of a series of slots. These slots are usually parameter-value-pairs that represent the information needed for the task. Figure 2.2 shows an example of such a frame in a train schedule information system [McT02]. The user is expected to fill these slots with appropriate values by answering questions of the system. After a slot is filled the system further needs to determine the next action that should be carried out. A common approach for this determination is state-based, i.e., for each filling status of a frame, there is a system state and each of these states is associated with some actions that should be triggered once this state is reached. Actions are, e.g., sending a query to a database if enough slots are filled or initiating a question if some slots still need to be filled. The advantage of the frame-based approach over the finite state-based approach can be demonstrated with the dialog example illustrated in Fig. 2.3. Here, the user provides more information than the system asked for. For efficiency reasons the system should be able to set the value for both the destination and the departure time in the current state. For a purely finite state-based approach this would be impossible because the system is only capable of receiving *the* piece of information that the system is prepared for. To process the departure time the system has to be in another pre-defined state which could be several transitions away. But in the frame-based approach the goal of the system is to fill slots which ensures optimal information extraction from a user utterance. This feature enables mixed-initiative dialog style, even if only in a relatively restricted manner. The frame-based approach (with different extensions and in different complexities) is very popular in dialog systems developed today and is implemented for a variety of domains: train schedule service [AOSS95, SdOB99], flight booking service [SP00], advertisement enquiry system [GMP+96], movie information service [CC99], travel booking agent [XR00], and so on.

Destination = London
Date = unknown
Departure time = unknown

Figure 2.2.: An example frame [McT02]

Sytem: Where are you travelling to?
User: I want to fly to London on Friday.

Figure 2.3.: A dialog example that can be better processed when using a frame based approach [McT02]

In comparison to finite state-based approach the frame-based approach enables greater flexibility for the user because it can handle extra information that the user provides. However, its basic idea is still to directly associate dialog states with task states and this approach is, therefore, also subject to the restrictions of the finite state-based approach.

2.1.3. The agent-based approach

Both the finite state-based and frame-based approaches follow a system- and domain-centered view of human-machine conversation: The system is an expert in the domain and possesses the knowledge that is needed to execute domain tasks. The user should provide enough information so that the system can initiate a task. In this context, the user plays a “subordinate” role since she is only an “information-provider” while the system decides what to do with the information. Starting from this view, these two approaches directly translate domain tasks into the dialog structure and mainly support system-led initiative distribution.

Human-machine conversation, however, can be viewed in a different way: in a way that we adopt when we consider conversations between humans. Since Grice [Gri75] first proposed the *cooperative principles* of human-human conversations the *collaborative nature* of conversation has been intensively studied and widely accepted. The basic idea is that dialog can be viewed as a collaboration between its participants who need to coordinate with each other during a conversation. Adopting this view to human-machine conversation means to view the machine also as an conversational agent that participates in the conversation in the same collaborative way as the human user. Based on this understanding, various new dialog modeling strategies have been developed which are categorized as *agent-based* approaches by McTear. Four of such strategies are briefly described in the following.

Using Theorem Proving

The Circuit-Fix-It Shop [SH94] is a dialog system that helps users to fix an electronic circuit. The system possess the complete, theoretic knowledge of how to do it, but has no sensory possibility to monitor and manipulate the state of the world. The user has this manipulating ability but may be novice in this field. Thus, the system and the user have to work together to solve the problem. The dialog evolves as a proof of the task completion: A task can be represented as a goal tree and the system invokes rules to prove the goal. Sometimes, the internal knowledge of the system is sufficient to prove a subgoal, but sometimes axioms of rules are missing that should be provided by the user, e.g., to physically connect two electrical connectors. In this case the system engages in a dialog to ask the user to do it. If the user can not provide the missing axiom, e.g., because the user does not know how to do a subtask, the system inserts a sub-step into the goal tree and first handles this new subgoal by teaching the user with its theoretic knowledge. After this problem is solved the system resumes the theorem proving process along the goal tree. The process of adding a new subgoal is illustrated in Fig. 2.4.

In this system the user and the system collaborate in the sense that both of them contribute to the dialog with their domain knowledge or domain-relevant abilities. In contrast to the finite state-based and the frame-based approach, the dialog participants can dynamically take initiatives since it is based on the status of the flexible, interruptible theorem prover instead of pre-defined task states.

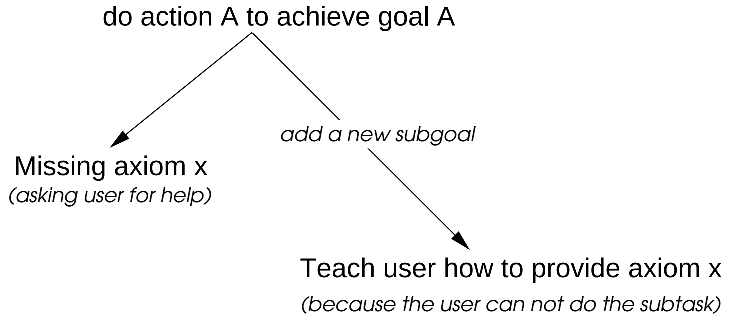

do action A to achieve goal A
add a new subgoal
Missing axiom x
(asking user for help)
Teach user how to provide axiom x
(because the user can not do the subtask)

Figure 2.4.: The process of adding new subgoal when using theorem proving [McT02]

The approach of theorem proving is also used by Sadek [SFC+96] to develop a telephone service dialog system. He stresses the rationality of intelligent behavior and proposes the *rationality principle* and the *cooperative principle*. The rationality principle states that an agent can not intend to bring about some proposition without intending and she will perform actions based on her intention to achieve the desired effect. This principle thus characterizes the agent's planning mechanism. The cooperative principle expresses agent's motivation to adopt her partner's intention whenever she has no reason not to do so. Sadek further proposes a communication theory [Sad94] which incorporates the rationality principles and communication act models in a formal way.

Plan-based Approach

In plan-based approaches utterances are viewed as actions that are performed to achieve some goal, e.g., to execute a task or to communicate information. In this context, utterances (both from the system and the user) are equivalent to *action operators* in the field of task planning and are often modeled as speech acts that consist of roles, preconditions, constraints, and effects. To achieve a goal, appropriate speech acts have to be chained together in the correct sequence similar to building a plan by putting correct sub-plans together. The dialog, thus, is governed by the planning mechanism that generates expectations what speech act, either from the system or from the user, is needed according to the plan status. In a train ticket purchasing example in [All95] the user is expected to produce a "MotivateByRequest" act in order to ask the system for the price of a ticket. Based on this user act the system needs to produce a "ConvinceByInform" act to inform the user of the price.

The early plan-based models had two main problems: Firstly, the mechanism heavily relies on the correct recognition of user intentions to locate user speech acts in the current system plan, which is not always easy. Secondly, the chaining of speech acts based on fulfillment of preconditions can become unmanageably complex in some cases. Litman and Allen [LA87], therefore, extended the basic model so that it can handle clarification questions which are urgently needed to cope with incorrect intention recognition results. Many alternative approaches were proposed based on this plan-based one. In the next paragraph one of these approaches will be introduced.

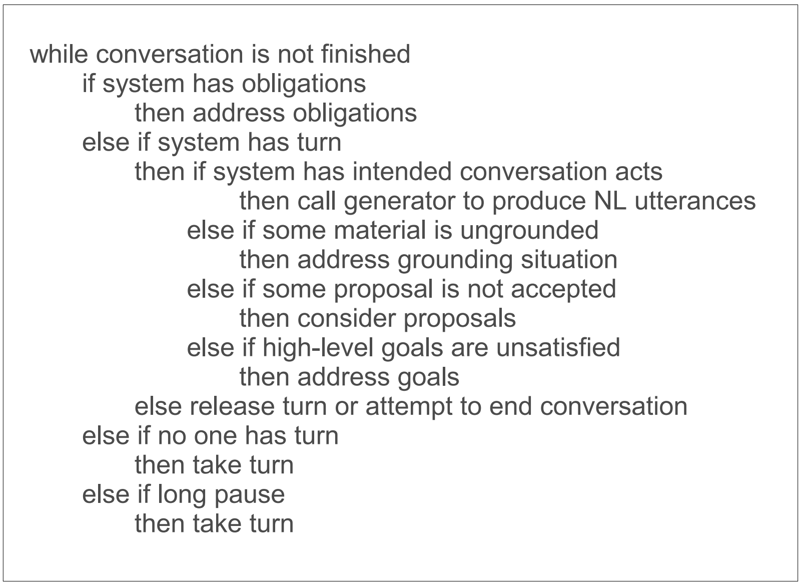
Collaborative Discourse Principles

COLLAGEN[RS97] is a tool kit for building collaborative interfaces and is based on the collaborative discourse principles. These principles derive from the theories on *SharedPlan* and the *focus of attention* of Grosz and Sidner[RS97, SKLL04]. SharedPlan is the formal representation of the mental aspects of collaboration participants: mutual beliefs about the goals, actions to be performed, capabilities, intentions and commitments. Focus of attention shifts during a collaboration, which is modeled by a focus stack of discourse segments. Each discourse segment is associated with a SharedPlan as the segment's purpose. During a conversation, COLLAGEN updates an agent's internal discourse state representation based on these principles. The discourse consists of the focus stack, history list (a record of top-level segments that have been popped off the stack), and the recipe tree (a concrete representation of some of the mutual beliefs in SharedPlans). COLLAGEN is one of the few sophisticated dialog modeling approaches that are actually implemented for robots (compare to section 2.4).

Conversational Agency

The TRAINS/TRIPS project [AS91, FAM96, FA98, ABD+01, AFS+05] is a long-term research project in which conversational collaborative planning systems for various domains have been developed. In their 1993 version of the system [Tra96] Traum proposed to view a dialog management system as a conversational agent that should be able to handle social attitudes (including mutual belief, shared plans, and obligations) and discourse context. During a dialog the system performs communicative acts (by generating speech output to the user), observes user's communicative acts, and maintains a model of its own belief, the belief of the user as well as the shared belief. Based on the mental knowledge, domain knowledge provided by other modules, and the discourse obligations the dialog management system makes decisions on system's next step.

The introduction of discourse obligations is the major advantage of this approach over the purely plan-based approaches. The focus of the plan-based approach is to recognize user intentions, infer her goals and adopt a goal to achieve the user's goal. This approach is based on the assumption of user's cooperativeness and can not explain why the user still needs to respond if she does not know the answer. The solution of the conversational agency is to distinguish domain-related intentions from socially or conversationally based obligations, e.g., if one asks a question the other is expected to at least signal her hearing. As illustrated in Fig. 2.5 the discourse management of TRAINS-93 also takes into account many important conversational issues like grounding and turn-taking. This algorithm produces a reactive-deliberative behavior of the dialog agency: On the one side, if the system has conversational obligations it will first address them rather than its own domain goal which means that the concern of the user is first considered; On the other side, if the user does not take turn the system will take this opportunity to address its own domain goals. Thus, the system can dynamically shift its focus from user-led initiative taking style (based on its own conversational obligations) to system-led initiative taking style (based on its high-level goals).



```
while conversation is not finished
    if system has obligations
        then address obligations
    else if system has turn
        then if system has intended conversation acts
                then call generator to produce NL utterances
            else if some material is ungrounded
                then address grounding situation
            else if some proposal is not accepted
                then consider proposals
            else if high-level goals are unsatisfied
                then address goals
        else release turn or attempt to end conversation
    else if no one has turn
        then take turn
    else if long pause
        then take turn
```

Figure 2.5.: Discourse actor algorithm in TRAINS-93 [Tra96]

Another important characteristic of Traum's model is the specification of "conversation acts" [Tra94]. The basic idea is that different types of actions are being performed during a dialog. Additionally to propositional-level actions such as speech-acts, other actions such as turn-taking and discourse segmentation also occur to maintain a coherent conversation. Therefore, he proposed a multi-layered representation of conversation acts including *turn-taking*, *grounding*, *core speech acts* and *argumentation acts* (later termed as forward and backward-looking acts [Tra97]). During a dialog the acts on different layers are updated which changes the state of this layer. This, in turn, changes the state of the overall dialog. Generalizing this principle, he and his colleagues developed the "*information state theory*" of dialog modeling [LT00, CL99] in the late 1990s. The basic idea of this theory can be summarized as follows: *A dialog has different aspects and each of these aspects can be represented as containing specific information. During a dialog, dialog moves modify this information and thus change the states of these aspects*. This theory represents his idea of dialog modeling in general and consists of

- a description of *informational components*, that is, aspects that need to be modeled, e.g., common ground, linguistic structures, obligations, beliefs etc,
- the *formal representation* of the above informational components,
- a set of *dialog moves* that triggers the update of the information state,
- a set of *update rules* that specifies how the dialog moves update the information state of the informational components and
- an *update strategy* that decides which rule to select given a group of applicable ones.

This principle of modeling dialog has been later extended by Traum to model multi-modal, multi-party dialog [TR02, SGH+04] which will be described in more detail in section 2.2.2.

Further, Traum proposed a computational theory of grounding [Tra94] which is one of the first implementable grounding models for dialog. Grounding is the process of establishing mutual understanding between dialog participants during a dialog and was first proposed in a systematical way by Clark[Cla92]. Traum improves Clark's model by representing grounding units, i.e., the dialog segment in which grounding takes place, with "discourse units" and modeling the dynamic process of grounding using a finite state-machine. Chapter 3 will provide a detailed account of this theory.

The approach of conversational agency separates conversational from domain tasks by also considering issues like grounding, turn taking, and obligations and stressing the role of discourse context. Given this feature, the approach of conversational agency distinguishes itself from the theorem proving and the plan-based approach, which equate the conversational collaboration with the domain task collaboration.

2.2. Handling multi-modality of dialog

Cognitive science and communication studies have provided the theoretical foundation for the multi-modal research: The cognitive science provides evidence for the cognitive association of language production in different modalities and communication studies document their observations of correlations between verbal and non-verbal behavior during face-to-face communication. The application of these ideas in computer science has been developing in two directions: One strand of research concentrates on the fusion mechanism of different modalities from the view of system architecture to improve the interaction efficiency in HCI. The other strand focuses on the development of multi-modal communication models that should account for natural and human-like interactions in the context of the embodied communication. This section presents the foundational work in these two strands though the multi-modal research in the context of embodied communication is discussed in more detail because of its greater relevance to the current work.

2.2.1. Multi-modal research with modality fusion and representation as focus

This strand of research started in 1980 as Bolt [Bol80] published his "Put That There" demonstration system which processes speech with pointing gestures to enable users to create and move objects on a 2-D large-screen display. Since the last decade the multi-modal research has been developing rapidly because of the progress made in its contributing technologies such as speech, gesture recognition and modality integration paradigms. To date, the most mature multi-modal applications in HCI are systems that combine speech and pen input [OC00] or speech and lip movements [BMP+00]. An overview of currently available systems that are already beyond the stage of prototype can be found in [OCW+00]. Relatively new in this field are the applications that enable vision-based technologies, such as the recognition of head position, gaze, posture, facial expression and manual gesture, as referred to as "perceptual user interface" by [TR00]. In contrast to traditional applications using active, intentional user input mode like speech, these interfaces can *unobtrusively* monitor user behavior and do not require explicit user commands. Such kind of technologies are, however, less reliable in interpreting user intention so that the development of "blended interface style", is becoming important, i.e., to combine passive (like often unconscious manual gesture) and active (like speech) human user input [Ovi03].

Since the research on multi-modal interfaces is largely enabled by the progress achieved in research on individual modality processing, this strand of multi-modal research mainly focuses on the fusion mechanism of different modalities and multi-modal meaning representation from the view of system architecture. Oviatt [Ovi03] summarizes the state-of-the-art approaches in this field. There are mainly two types of multi-modal fusion, "feature-level" and "semantic-level" fusion. Feature-level fusion integrates signals at the feature level and is suitable for combining modalities that are temporally closely synchronized, e.g., speech and lip movement. Semantic-level fusion is typically applied for modalities without close temporal coupling like speech and manual gesture because they provide different, but complementary information that is usually fused on the utterance level. To fuse meanings derived from different modalities Vo&Wood [VW96] and Pavlovic&Huang [PH98] propose the *frame-based* integration strategy of recursively matching and merging attribute-value data structures. An alternative approach is the *unification-based* integration that unifies feature-structures and is inspired by computational linguistics [Car92]. This approach is considered as more suitable for complex multi-modal meaning integration.

Researchers in the SmartKom project [Wah03, PAB03] have developed a user interface to help users operate a phone, select media content and navigate in three different scenarios with a life-like character Smartakus[1]. The central issue of this application is the resolution of multi-modal references. For this purpose, a three-layered multi-modal discourse representation (Fig. 2.6) is proposed: The *modality layer* consists of linguistic, visual and gestural objects representing concrete realization of a referential object in the real world; Objects on the *discourse layer* represents concepts which potentially serve as referent for referring expressions; The *domain layer* links the discourse layer objects with the system's ontology representing domain tasks and objects.

[1] Although this character can also generate gesture, facial expressions, etc, it is not the focus of the project. Therefore, this system is introduced in this section instead of the next section.

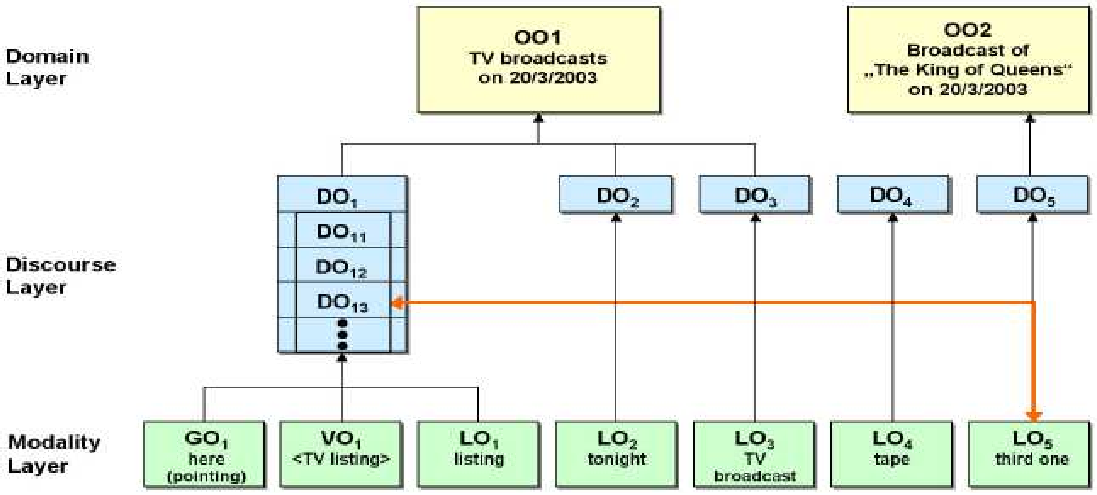


Figure 2.6.: SmartKom Multi-modal discourse representation ([Wah03])

2.2.2. Multi-modal research with communication modeling as focus

In the context of embodied communication, the issue of multi-modality has been intensively studied for the development of conversational virtual agents[2]. The following two subsections discuss two different strands of dialog modeling research in this field, as illustrated in Fig 2.7. The first one discusses approaches that focus on the optimal modeling of the *relationship* between multi-modal, *individual* dialog contributions and thus hold a horizontal view of multi-modal dialog (the blue field in Fig 2.7). The second subsection concerns approaches that vertically view dialog and focus on the *discourse modeling* of the *entire dialog* with multi-modality as different forms of contribution (the pink field in Fig 2.7).

Multi-modal dialog: a horizontal view

The focus of this strand is the animation of synchronized, multi-modal behaviors for virtual agents as imitation of natural human behaviors. Relevant observations made in human communication studies and the theories derived from these observations constitute the theoretical basis for the animation.

Communication studies on the relationship between verbal and non-verbal communication behaviors of human go back to the 1960s. Early works were based on a *channel summation model* which assumed that the verbal and non-verbal behaviors convey generally different kinds of messages and the total meaning of these messages can be derived from the frequency, intensity and

[2] A common term of such agents is “embodied conversational agents” [CBCV00, NIL00, RDN02]. Although conversation within the HRI context is also an embodied conversation, this term is only commonly used to refer to *virtual* agents in HCI field. To emphasize the difference between the interaction with a virtual embodied conversational agent and with a robot, the term “virtual agents” is used in the current work.

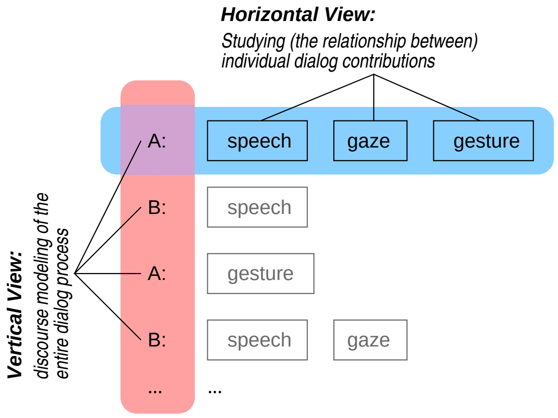

Horizontal View:
Studying (the relationship between) individual dialog contributions
Vertical View:
discourse modeling of the entire dialog process
A:
speech
gaze
gesture
B:
speech
A:
gesture
B:
speech
gaze
...
...

Figure 2.7.: The two research foci of multi-modal dialog modeling, illustrated with an excerpt of an abstract dialog between participants A and B

relative weighting of acts summated across channels [JL02]. Subsequent research rejected this additive manner of meaning building, e.g., Hegstrom [Heg79] proposed that the total meaning depends on the particular combination of messages conveyed in different channels and Duncan [Dun72] claims that the impact of behaviors is derived from their sequential or simultaneous relationship, or both. The current focus of multi-modality research are the mutual or co-active influences between different modalities. One of the foundational works that contributes to this trend was the early discovery of Condon [CO71] about *self-synchrony* and *interactional synchrony*. By analyzing human conversation on films frame by frame, he found out that various parts of the human body move in time with each other as well as with the articulation of her speech (self-synchrony), and the listener's behavior is also organized self-synchronisely following similar pattern as the speaker (interactional synchrony).

These earlier works are still inspiring researchers today, e.g., Cassell's work on conversational virtual agents. In her early work "animated conversation" [CPB+94] she developed a system that automatically generates context-appropriate gestures, facial expressions and intonational patterns for virtual agents. In order to avoid issues related to human activity detection and recognition the interaction takes place between two virtual agents, who play the role of a bank teller and a client, respectively. In the interaction, the client asks the bank teller to help him obtain 50 Dollars with a check. The focus of this work was the generation of non-verbal behaviors.

Cassell's colleague Thorisson [Thó96] stresses turn-taking in speaker-listener relationship and proposes a multi-modal dialog model for virtual agents that view the interaction as a *three-layered feedback loop* (see. Fig. 2.8). The bottom layer deals with *reactive* conversational actions like looking away when the speaker believes it is her turn; the middle layer is concerned with processes that have direct reference to the *dialog process*, e.g., utterances like "I'm trying to remember...". On the top layer the *content* or the topic of the conversation is processed. This model was implemented for Gandalf [Thó97], a conversational virtual agent who shows users the model of a solar system.

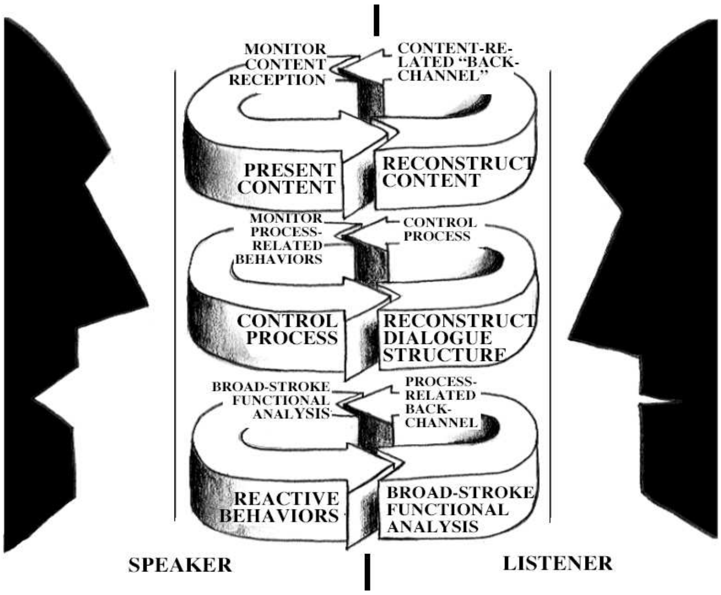

MONITOR CONTENT RECEPTION
CONTENT-RELATED "BACK-CHANNEL"
PRESENT CONTENT
RECONSTRUCT CONTENT
MONITOR PROCESS-RELATED BEHAVIORS
CONTROL PROCESS
CONTROL PROCESS
RECONSTRUCT DIALOGUE STRUCTURE
BROAD-STROKE FUNCTIONAL ANALYSIS
PROCESS-RELATED BACK-CHANNEL
REACTIVE BEHAVIORS
BROAD-STROKE FUNCTIONAL ANALYSIS
SPEAKER
LISTENER

Figure 2.8.: Three layered feedback loop in multi-modal dialog [Thó99]

Based on Thorisson's and her own earlier work, Cassel and her colleagues proposed a generic architecture for conversational virtual agents and implemented it for REA [CBCV00]. REA is a virtual real estate salesperson who shows users around virtual properties and attempts to sell them a house. This architecture represents the central idea of Cassell's FMTB conversational framework (function, modalities, timing, behaviors): Multiple (*interactional* and *propositional*) communicative goals are conveyed by conversational *functions* that are expressed by conversational *behaviors* in one or several modalities. In this model, the interactional goals regulate the state of the conversation, e.g., establishing contact with the user or releasing turn, while the propositional goals are driven by the needs of discourse.

In the architecture illustrated in Fig. 2.9, the Input Manager collects input from all modalities. Data that require instant reaction are categorized as requiring Hardwired Reaction and are directly sent to the Action Scheduler, which generates multi-modal output in a synchronized manner. Input data that need deliberate discourse processing are forwarded to the deliberative module for interpretation and response generation. Here, interactional and propositional information is processed by two different modules. The processing results of these two modules are conversational functions that are subsequently converted into different conversational behaviors by another module.

Multi-modal dialog: a vertical view

In the above systems, the user and the virtual agent have access to the same objects or environment (the solar system model in Gandalf and the houses to be sold in REA), but they do not *cohabit* the same environment. Virtual agents that do share the same physical environment as the

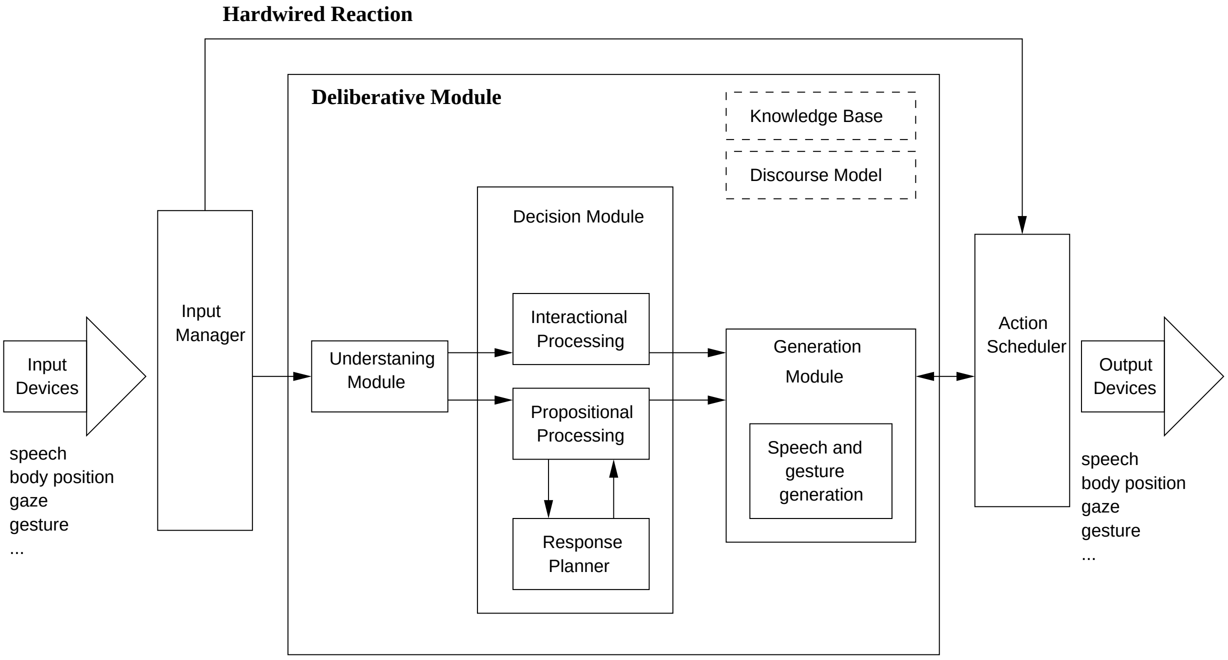


Figure 2.9.: A generic architecture for embodied conversational agents [Cas00]

human user raise additional issues for interaction modeling. For example, the shared environment is usually a relatively large one so that the user can move around to interact with various objects or agents. This point raises the issue of attention, i.e., the user is not necessarily attending to the agent as it wants to address her. The realization of such systems requires more robust and flexible dialog management strategies to account for the variability of the shared physical environment.

Steve [RJ00] is such an agent and acts as instructor and teammate for human students in naval operating procedures. The dialog management strategy of this system is not yet sophisticated: It operates by selecting his next action from a repertoire of behavioral primitives, e.g., speaking, moving to an object, pointing at an object, offering turn, and the system represents the dialog context using a set of rules.

Traum and his colleagues have been working on the Mission Rehearsal Exercise project (MRE) [TR02, SGH+04]. Within this project interactive virtual humans in a peacekeeping scenario for training purposes are created. The dialog model of this system is built on the information state theory as described in section 2.1 on page 15. To fulfill additional requirements of virtual reality dialog systems including the issue of multi-modality, attentional issues and the issue of multi-party dialog [TR02], he extended the four-layers of conversation acts applied in TRAINS-93 [Tra94] to five main layers and the layer of Conversation further contains six sub-layers (see Fig. 2.10). In accordance with the information state theory, each layer includes an information state that is to be changed by dialog moves (termed as “dialog acts” in the MRE project). This model accounts for multi-modality of conversation by defining dialog acts that

- Contact
- Attention
- Conversation
  - Participant
  - Turn
  - Initiative
  - Grounding
  - Topic
  - Rhetorical
- Social commitments (obligations)
- Negotiation

Figure 2.10.: Multi-party, multi-conversation dialog layers [TR02]

must or can be realized using a specific modality or a combination of multiple modalities. To illustrate the principle of Traum's multi-modal dialog model the following paragraph presents some details of its *contact* and *attention* layer. These two layers have been added to his original dialog models [Tra94, LT00] to fulfill the requirements of multi-modal, multi-party dialog in a virtual world.

The contact layer concerns whether and how other individuals can be accessible for communication. Modalities that can be activated for this purpose include visual, radio and voice (e.g., shout). Actions that can influence the state of this layer are *make-contact* (e.g., by walking into the view or earshot) and *break-contact* (e.g., by moving behind something). These two actions change the state of this layer which indicates whether the interaction partner is in contact or not. For example, in one MRE scenario, as a lieutenant drives up to the sergeant and walks out of his vehicle, he becomes visible to the sergeant with whom the contact is thus established. The "in contact" state on this layer is the prerequisite for attention. The attention layer concerns objects or processes that agents attend to. Actions affecting this layer can be actions that an agent performs concerning its own attention (*give-* and *withdraw-attention*) or those related to the attention of other agents (*request-*, *release-* and *direct-attention*). As on the contact layer, these actions can be performed in a multi-modal way as well. The operation on other layers follows the similar principle as on these two layers. During a dialog, one agent establishes contact with the other agent by signaling her communication intention using an appropriate modality. Then, the agent being addressed demonstrates her attention to the dialog initiator also using certain modality. Thus, the basis for a dialog is created so that the dialog participants can go on providing dialog acts that change the states on other layers of the model. The resulting multi-modal behaviors are animated and demonstrated on the screen.

Nakano et al. [NRSC03] adopted the above information-state based discourse modeling approach for their virtual, conversational kiosk MACK [CSB+02], whereas this work focuses on the grounding function of non-verbal behaviors, particularly gaze.

2.3. Evaluation of interaction systems

As mentioned earlier in this chapter, the issues of spoken dialog modeling and multi-modality management have been the focus of two different research traditions. Accordingly, the evaluation effort was also made separately from each other. In the following, the evaluation works in the field of dialog management systems and virtual agents are presented in sections 2.3.1 and 2.3.2.

2.3.1. Evaluating dialog systems

The majority of the effort put into the development of evaluation metrics concerns the information retrieval domain. One of the first attempts to evaluate the performance of dialog systems is based on the notion of reference answer [HDM+90]: The responses that a dialog system generates are compared with a pre-defined key of minimum and maximum reference answers and the performance of the system is indicated by the proportion of responses that match the key. This approach can only account for responses generated using *one* dialog strategy. Subsequent researchers have proposed various metrics for the evaluation of dialog strategies by carrying out real system tests. Especially the metrics proposed from the SUNDIAL [Pec93] project are worth mentioning [SF93]:

- *Transaction success*: This metric measures how successful the system has been in providing users with the requested information.
- *Number of turns*: This is a measure of the duration of the dialog in terms of how many turns were needed to complete the transaction. An alternative measure is the time taken to complete the transaction. These measures are also indicators for Transaction Success or User Acceptance.
- *Correction rate*: This is a measure of the proportion of turns in a dialog that are concerned with clarifications and corrections (of speech recognition, understanding and conception). A high degree of Correction Rate indicates high costs in terms of User Acceptance.
- *Contextual Appropriateness of utterances*: This is a measure of the extent to which the system provides appropriate responses. The metric can be divided into a number of values, such as total failure, appropriate, inappropriate, appropriate/inappropriate (when the evaluator is in doubt), and incomprehensible.

Danielli and Gerbino [DG95] propose a more qualitative metric the *implicit recovery,* which captures the ability of the dialog management system to recover from partial or total failure of speech recognition and speech understanding. The basic idea is that the interaction time can be shorter if a system is able to reason about user's actual intention and automatically recovers from the failure of the speech recognition and understanding.

The metrics presented so far often have relations to each other. For example, shorter transactions enabled by implicit recovery are often realized at the cost of lower robustness, i.e., possibly lower transaction success, and it is difficult to determine which aspect is more critical for the performance. Besides, environmental factors (e.g., noise) and task factors (e.g., database size) also have influence on the performance which can not be covered by the metrics above. To address these limitations Walker [WLKA97] proposes a general framework for evaluating dialog systems called PARADISE. This framework combines various performance measures such as transaction success, user satisfaction, and dialog cost into a single performance evaluation function and enables performance to be calculated for subdialogs as well as for the complete dialog. Figure 2.11 illustrates the basic structure of PARADISE: Performance is modeled as a weighted function of a task-based success measure (transaction success) and dialog-based cost measures. The transaction success is calculated based on a general task representation of a so-called *attribute value matrix* (AVM): It is ordered pairs of attributes and values which represent all the possible task information to be exchanged between the system and the user. For example, an attribute is *depart-city* and the values are *London*, *Paris* and *Berlin*. Each subdialog reflects (a collection of) task information which is independent of the dialog strategy involved in this subdialog. Based on this representation, data is collected in a confusion matrix according to whether the values have been recognized correctly or not. The Kappa coefficient, κ, is calculated for each attribute which indicates how well the system has achieved the information requirements of a particular task. To calculate the dialog costs the collected dialog data is hand-tagged with quantitative and qualitative tags, e.g., which subdialog is a repair dialog and how many repairs have occurred in the interaction. The overall user satisfaction is calculated as a weighted function of the transaction success and dialog cost. The PARADISE evaluation framework incorporates and enhances previous evaluation measures by (1) separating *what* tasks need to be achieved from *how* they are achieved and (2) using a decision-theoretic framework to specify the relative contributions of various factors to a system's overall performance. This framework can account for the measurement of different dialog strategies and has been used for large cross-system evaluation in information retrieval domain, [WAB+01, WRP+02].

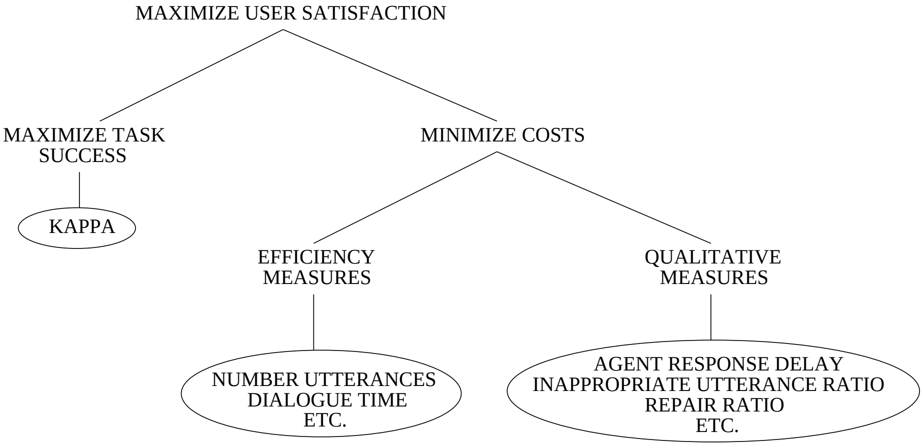


Figure 2.11.: PARADISE's structure of objectives for spoken dialog performance [WLKA97]

2.3.2. Evaluating conversational virtual agents

In the field of conversational virtual agents, it is difficult to directly use many of the above metrics to evaluate the dialog capability of an agent because of the multi-modality of the interaction. Many information retrieval systems are purely speech based and only some allow pen or keyboard as alternative input channel. Even in these systems the modalities do not co-carry meanings as it is the case in most embodied interaction. This means, the responsibility for transaction failure, for instance, is clearer in this domain than in virtual agent domain. Here, other system modules are usually needed to interpret meanings carried by non-verbal modalities like gestures and postures. Besides, these additional image processing modules often also claim plenty of computation time which negatively influences the dialog duration, too. If these modules fail to generate an interpretation of user input, the spoken dialog system certainly has to initiate clarification questions which constitute system repair. Thus, evaluation metrics like transaction success, dialog duration, correction rate etcetera in their original sense can not directly reveal the efficiency of the employed dialog strategies, i.e., the efficiency of the implemented dialog model. Besides, there are additional aspects that should be evaluated for embodied conversation, e.g., the naturalness of the generated multi-modal behavior.

The majority of the evaluation work done on virtual agents focuses on the evaluation of, instead of the interaction capability, likeness and user preference in terms of agent's appearance or non-verbal behavior. Table 2.1 provides an overview of some evaluation work done in the field of virtual agents [RDN02]. Developers of more sophisticated virtual agent systems as those discussed in section 2.2.2 adopt composed evaluation strategies with *system-specific metrics*. The virtual agent Gandalf (page 19) was evaluated in 3 steps [Thó96]. The first step was the comparison of the performance of Gandalf with the Model Human Processor [CMN86], which is a model designed to predict human performance and reaction time. The goal is to find out whether the behaviors of Gandalf is similar to that of the human as predicted by the Human Processor. The second evaluation step was the conduction of a comparative user study with three agents with different behaviors to test whether the developed characteristics are important. The final step was observations of the interaction by the author himself. The evaluation plan of virtual agent REA (page 19) is to use three different criteria [CBCV00]. The first criterion is the amount of possible lacunae in the theory that would be pointed out by their implementation. The second criterion is the amount of aspects of the proposed model that can not be translated into the system behavior. The last criterion are metrics in comparative Wizard of Oz studies. The goals of these studies are user perception of the system and task execution ratio. The Mission Rehearsal Exercise system by Traum [TR02, SGH+04] (page 21) was evaluated in the aspects of [TRS04] (1) user satisfaction (by conducting user studies), (2) subjective task completion ratio (a ratio of all tasks the trainee attempted) and objective task completion ratio (the ratio of only those tasks included in the system's domain), (3) recognition rate, and (4) response appropriateness. For (2) and (4) Traum also proposes a discourse structure coding scheme, the so-called *IU-coding* [NT99] and an appropriateness coding scheme [TRS04].

Ref	Changed parameter	Evaluated parameter	Method of data collection	Subjects	Application	Findings
[NIL00]	personality: introvert/extrovert (posture); ethnicity (by look); (in)consistent verbal/nonverbal	Trust, liking	questionnaire	40 Korean students, 40 students (extrovert/introvert)	application independent ("item selection" arguing)	more trust in extrovert and identical ethnicity virtual agents
[CT99]	envelope emotional feedback	ease of use, efficiency, lifelikeness	Survey + performance data analysis	12 novice comp. users, native English speaker	Information provider about the solar system	envelope is more important than emotional feedback
[CCD00]	eye-gaze	eye response by user, turn taking	analysis of eye pattern of user	20, CS staff	casual chat	For lester avatar with eye-gaze, users respond with eye-gaze
[LCK+97]	pedagogical agent with different modalities	effectivity liking entertainment etc. (18 aspects)	performance test Data analysis	100 secondary school students/novice-expert student	plants	lifelike agent has positive effect on learning: performance & experience depending on expertise of student
[KM96]	smiley, dog (realistic/cartoon), man, woman, no face	Involvement, likability	usage data registration + online questionnaire via Internet	157 out of 1000+ users, mostly men	poker game	face is engaging, likable and comfortable, all faces were attributed with intelligence, realistic ones the most
[MAMJ01]	3d woman/man, formal/informal, appl. domain	aspects of liking trust	questionnaires	36 subjects	banking/cinema/ flight	Trust less in case of banking appl, dress requires according to appl.
[MSJW00]	video/talking head/still with moving lips/voice	liking	questionnaires + focus group	–	textile e-retail	Video liked best, talking head least! voice only was liked
[BC01]	smalltalk	trust liking	questionnaires + analysis of behavior	18 students	house sail	smalltalk induces trust with extrovert subjects
[WSS94]	no face neutral face/stern face	liking effectivity	questionnaire	49 adults from CS research environment	filling in the questionnaire	voice only least liked & inefficient; neutral face liked most, stern face was efficient

Table 2.1.: An overview of some evaluation work in virtual agent research

As can be seen, dialog modeling and evaluation for virtual agents cover much more aspects than for most traditional dialog applications. Not only do they have to account for the issue of multimodality, they also have to take care of the "pre-dialog" phase of an interaction. As presented in Table 1.1 on page 5, interaction modeling for HRI is even more challenging, especially given the complexity of the overall robot system and the real environments. As shown below, many HRI applications, therefore, side-step the problem by adopting quite simple modeling approaches and few of them was ever systematically evaluated.

2.4. State-of-the-art in HRI

Service robots for naive users that have been intensively studied are tour guide robots. The interaction with such robots is often carried out via asymmetric modalities, i.e., users have to push buttons or click on a touch screen while the robot can provide feedback via speech, facial expression and display, e.g., robots Sage [NBG+99] or RoboX [TPJ+02]. Some others do not have a real dialog management system, but a command matching mechanism to enable basic interaction via speech. For example, office guide robot Polly [Hor93, Hor96] and museum guide robot Jinny [KCH+04]. Similar dialog management systems are documented for robots MAIA [ACC+94], RHINO [BCF+98], Minerva [TBB+00], MOPS [TVS01], and Perses [BWK+03]. Also for some personal robots like elderly care robot Flo [RBF+00] or robot pet AIBO [Kap00] simple key word spotting and command matching techniques are employed to realize interaction.

Many more advanced service robots possess “real” dialog management system. The finite state-based approach as discussed in section 2.1 are often adopted, e.g., the office service robots Jijo-2 [FAM98, MAF+99, MhA+00], Cero [GE01], and HERMES [BG02], hospital service robot Hygeiorobot [SAS01], intelligent environment interface robot Lino [KPC+03], humanoid robots Qrio [AS05] and Alpha [BFJ+05] for interaction research. Even the former version of the dialog management system on our own robot BIRON [TLWF04] was finite state-based.

Several other projects try to adopt different strategies to enhance the verbal capabilities of their robots: The autonomous wheelchair robot Rolland [SB05] extends the finite state-based approach by explicitely addressing issues like question in discussion, belief, and abstract interface to knowledge. The elderly guide robot Pearl [MPR+02] uses a probabilistic approach to calculate the uncertainty of the speech recognition results and can generate clarification questions if appropriate. Single task navigation robot Godot [TBC+02] adopts the information state approach as discussed in section 2.1 and subsection 2.2.2. The dialog building toolkit COLLAGEN (section 2.1.3 was implemented for the penguin robot Mel that is supposed to engage human visitors in a conversation while demonstrating a so-called IGlassWare table.

Since the majority of service robot dialog systems are finite state-based, most of them have limited capabilities and can not account for multi-modality of conversation *within* the dialog system. For example, the generation of conversational facial expressions is widely realized for robots. However, these non-verbal feedback capabilities are not controlled by the dialog management system, but an independent robot subsystem. In such robot systems, the control of the interaction behavior lies in the overall robot control system instead of the dialog management system. And it is similar with multi-modal input, e.g., Qrio [AS05] has a two-layered architecture with one layer processing non-verbal behavior and the other layer the verbal behavior.

In the field of social robotics, research focuses on the realization of sophisticated, *individual* social behaviors such as emotion [Bre00], joint attention [Nag04] and spatial perspective taking [HTHS04]. However, researchers in this field do not attempt to develop complete interaction framworks which account for multi-modality and discourse management etc.

2.5. The adopted approach

On view of the many dialog modeling approaches, a decision has to be made as to which approach should be adopted to model multi-modal interaction for a robot companion. Recall that finite state-based and frame-based approaches follow a domain- and system-centered view: the system is an expert in the domain and the user plays a "subordinate" role. The user is only a "information-provider" while the system decides what to do with the information. These two approaches directly translate domain tasks into dialog structure and mainly support system-led initiative distribution. The HRI requirements of domain-independence and mixed-initiative dialog style for a robot companion, as discussed in section 1.1.2, can thus hardly be fulfilled with these two approaches. Agent-based approaches are more promising because they view dialog as a cooperation between a human user and a system, which matches the human-robot relationship in HRI better. However, within this category, theorem proving, plan-based approach and the approaches based on collaborative discourse principle also rely on detailed domain task representations to perform dialog planning and are less suitable. In the remaining approach, the conversational agency, a grounding scheme is proposed that describes general laws governing dialog, independently of the domain, and is flexible in terms of initiative distribution. A grounding-based dialog model is thus adopted for the current work. Given the broad responsibilities of an interaction management system for a robot companion, the issue of multi-modality is viewed vertically (compare to Fig. 2.7 on page 19). More specifically, the general grounding concept is generated with the ability to directly handle multi-modality, as will be shown in the next chapter.

Concerning the evaluation of the interaction system for a robot companion, task success-centered metrics, which are popular in the information retrieval domain, are not appropriate. The reason is that it is difficult to *clearly attribute interaction failure to one or more system modules*. Additionally to the issue of multi-modality, which is also a challenge for evaluating virtual agents, a mobile robot is also confronted with the problem of (frequent) signal processing errors and their consequences. Therefore, in the current work, composed evaluation metrics are adopted in a system-specific way, as often performed in the field of virtual agents.

2.6. Summary

This chapter presented relevant works in the fields of dialog modeling and multi-modality research. The most common dialog modeling approaches were classified into three categories: finite state-based, frame-based, and agent-based approach. These approaches were discussed with respect to their major features, advantages and disadvantages. The works in the multi-modality research were categorized according to their foci: either modality fusion and representation or communication modeling. Within the category of communication modeling, the work of Cassell and Traum were discussed in more detail because they are representatives for two major strands: either emphasizing the relationship between individual multi-modal contributions during a conversation (Cassell) or focusing on the the discourse modeling of the *entire* process of multi-modal dialog (Traum). Concerning the evaluation techniques in interaction modeling, task success ratio and human-alikeness are the dominating evaluation metrics in information retrieval and virtual agents domain. Compared to the relatively advanced technologies in HCI, less sophisticated approaches commonly adopted in HRI were also discussed.

The conclusion of this chapter is that grounding is a promising concept for the interaction management system for a robot companion because it models dialog by addressing laws governing dialog in general and is sufficiently flexible. Next chapter proposes a computational model of multi-modal grounding which improves and extends existing works to fulfill interaction requirements for a robot companion.

3. A computational model of multi-modal grounding

This chapter proposes a computational model of multi-modal grounding to address the requirements for an interaction management system in HRI (see Table 1.1 on page 5). In comparison to existing works, this model [LWS06, LW07] possesses two novel aspects. The first one is the improvement of the grounding mechanism itself. Combining advantages of three existing models, the proposed grounding mechanism is based on a push-down automaton and is able to avoid many of the inherent problems of the existing models. The second novel aspect is the extension of the grounding model, which is primarily used to address uni-modal dialog in other works, with the capability of handling multi-modality. This aspect enables the grounding model to cover more dimensions of face-to-face interactions in general. In the rest of the thesis, this model is referred to as MMPDA (multi-modal push-down automaton) model.

This chapter is organized as follows: Section 3.1 presents three existing grounding models. Motivated by these works, section 3.2 discusses the new MMPDA model in detail. Finally, section 3.3 presents an evaluation of the model using dialog examples from the literature and summarizes the advantages and disadvantages of the MMPDA model.

3.1. Existing grounding models

The term "common ground" in the sense of information and/or beliefs that dialog participants commonly share was introduced as early as in the 1970s. Karttunen and Peters [KP75] and Stalnaker [Sta78] pointed out that dialog participants can not successfully talk to each other without sharing mutual knowledge and beliefs based on what has been said in the dialog. Since then, the process of achieving common ground, the so-called grounding process, has been viewed as an important mechanism that regulates dialog. Researchers from different disciplines have proposed various grounding models to "explain" human dialog behaviors and to realize natural dialog for computer applications. In this section three of the most influential models will be discussed with respect to their main structure, advantages, disadvantages and implementability.

3.1.1. Clark's contribution model of grounding

The *contribution model* of Clark [Cla92] is one of the first grounding models. He views dialog from the standpoint of a third person that does not actively participate in the conversation. The following paragraphs summarize the basic ideas of this model.

Dialog Contributions: Dialog consists of *Contributions* initiated by dialog participants. A Contribution is a participatory act and involves (1) the individual act of the speaker to contribute to the discourse, (2) the individual act of the listener to register what the speaker said and (3) the collective act of both to add what the speaker meant into their common ground. Conceptually, the process of contributing divides into two phases:

- Presentation Phase: A presents an utterance for B to consider.
- Acceptance Phase: B accepts A's utterance by giving evidence for her understanding.

Once B provides evidence for understanding, i.e., B shows Acceptance, both A and B will each believe that B understands what A meant and will add it into their joint common ground.

The recursive Acceptance: Every signal that one dialog participant directs to the other dialog participant is presented for her to consider. Therefore, every utterance in a dialog belongs to the Presentation phase of some Contribution. This is to say, that also each Acceptance is a Presentation because it also needs to be considered by the other dialog participant and grounding is thus a *recursive* process. In the dialog example in Fig. 3.1, B accepts A's Presentation by repeating it. Then A accepts this Acceptance by saying “yes” and so on.

The strength of evidence: Clark states that Acceptance, i.e., evidence of understanding, can be of different strengths. This strength seems to be determined by the fact of *how explicitly the evidence demonstrates understanding*. Generally, the strength of an Acceptance depends on the complexity and purpose of the Presentation. Clark identifies five types of evidence which are listed here in order from the weakest to the strongest: (1) continued attention, (2) initiation of the relevant next turn, (3) acknowledgment (e.g., saying "uh" or "yeah"), (4) demonstration (the listener demonstrates all or part of what he has understood), and (5) display (the listener displays verbatim all or part of A's Presentation).

The introduction of the notion of strength is an important part of the model because it helps to solve the problem that is created by recursive Acceptance. Viewing the grounding process as recursive means that even the last utterance of the conversation requires an Acceptance to be grounded which results in an infinite loop of Presentation and Acceptance in each dialog. To solve this problem, Clark proposes the *strength of evidence principle*:

> The participants expect that, if evidence e_0 is needed for accepting Presentation u_0, and e_1 for accepting the Presentation of e_0, then e_1 will be weaker than e_0.

Given this principle, the upshot of every Acceptance phase should end in the weakest form of Acceptance, i.e., either in type 1 or 2. The dialog example in Fig. 3.1 displays this process. Here, A informs B of the book identification number *F six two* and B accepts A's first utterance by displaying it verbatim which is the Acceptance type 5. Then A accepts B's acceptance by saying *yes* which is an acknowledgment (Acceptance type 3). In the end, B accepts A's acknowledgment by initiating a new contribution (Acceptance type 2).

A: F. six two
B: F six two
A: yes
B: Thanks very much

Figure 3.1.: A dialog example [Cla92]

Embedded Contribution: This term is introduced to cope with the problem that dialog participants sometimes have understanding problems and have to initiate repairs. The Contribution that the listener's "negative Acceptance" (such as "I beg your pardon?") initiates is subordinate to the initial Presentation of the speaker. This is to say, that the Acceptance of one Presentation can contain other complex Contributions. Such an example is illustrated in Fig 3.2. Here, Contribution C_2 and C_3 both belong to the repair effort of the dialog participants and are subordinate to the Contribution C_1.

The discourse as contribution tree: Clark's contribution tree reflects the ideas above. As demonstrated in the contribution tree in Fig 3.2, each Contribution has a Presentation phase as well as an Acceptance phase and each utterance belongs to the Presentation phase of some Contribution. Contributions are ultimately completed by either Acceptance type one or two (i.e., either continued attention or initiation of the relevant next turn). As a rule of thumb, a Contribution belongs to the Acceptance phase of a previous Contribution only if it directly addresses the hearing or understanding of the previous Presentation.

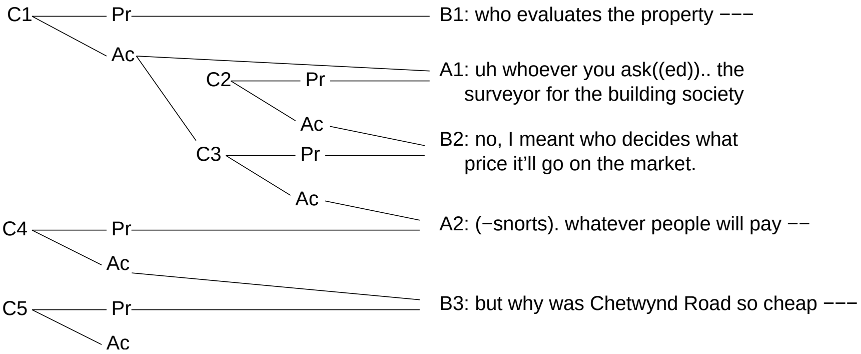

C1
Pr
Ac
B1: who evaluates the property −−−
C2
Pr
Ac
A1: uh whoever you ask((ed)).. the surveyor for the building society
C3
Pr
Ac
B2: no, I meant who decides what price it'll go on the market.
C4
Pr
Ac
A2: (−snorts). whatever people will pay −−
C5
Pr
Ac
B3: but why was Chetwynd Road so cheap −−−

Figure 3.2.: An example of dialog discourse as contribution tree [Cla92] (C = Contribution, Pr = Presentation and Ac = Acceptance).

Clark's contribution model is well-established and models the grounding process in an explicit and direct way. However, close examination of this first grounding model reveals several logical problems which have severe consequences for its implementability.

The first problem is the recursivity of Acceptance. Traum [Tra94] points out that if the Acceptance of the Presentation Pre_1, say Acc_1, also has to be accepted by another one, say Acc_2, then it is not clear whether the Acc_1 is really qualified to complete the grounding process of Pre_1 before Acc_2 is available. If no, then Acc_2 can not be qualified for any accepting function for Acc_1 either because it has to wait for Acc_3, and so on. Obviously, this will result in an infinite loop which implies that no utterance can be ever grounded. However, if Acc_1 can fulfill its accepting function without Acc_2, as Clark suggests, there seems to be no reason why it should be accepted by Acc_2 at all since the initiator of Pre_1 should already be satisfied with Acc_1 and does not need further Acceptance. It is, therefore, more reasonable to view Acc_2 as a pure Presentation without any accepting function, which means that there are utterances in the model (in this case the Acc_1) that do not need to be accepted at all. This is, however, contradict to the principle of recursive Acceptance. This logical problem makes it difficult to implement this model since it is not clear when the grounding process of an utterance is ever completed.

The second critical issue of Clark's model is the unclear relationship between the dialog regulation power of grounding and the role of domain task related motivations of dialog participants. This unclearance affects the conditions for initiating the next Contribution. On the one hand, one should follow the strength of evidence principle to generate an Acceptance that is weaker than the previous one. On the other hand, Clark also states that the Acceptance type is determined by the complexity and purpose of the Presentation. Thus, it is not clear whether a dialog participant initiates the next turn (Acceptance type two) because she is expected to provide weaker evidence of understanding than earlier or because she is motivated by the current dialog purpose. The contribution model thus lacks a specification of the interplay between the meta-process of grounding and the domain. This point is crucial for an implementation since the majority of computer applications is created to solve domain problems rather than to prove the correctness of dialog theories.

Last but not least, the role of non-verbal conversational behaviors is not clear in Clark's model. Although Clark does mention that non-verbal behavior can also serve as Presentation and Acceptance (e.g., the continued attention is a type of Acceptance), his contribution model does not provide any account for it. For example, questions like how are non-verbal behaviors embedded in the dialog, how to handle simultaneous verbal and non-verbal contributions etc are left unanswered.

As a whole, Clark's contribution model attempts to *describe dialog afterwards* instead of *predicting it beforehand*. Some of its logical problems severely affect the implementability of this model. Subsequent researchers have proposed various modifications for this model to address these problems. In the following, two of such models will be discussed: The finite state-based model of Traum and the exchange model of Cahn and Brennan.

- *Initiate*: initiates a DU
- *Continue*: continues a previous act performed by the same speaker
- *Acknowledge*: shows understanding of a previous utterance initiated by the other party (comparable to the Acceptance of Clark)
- *Repair*: changes the content of the current DU
- *ReqRepair*: asks for a repair by the other party
- *ReqAck*: attempts to get the other agent to acknowledge the previous utterance (creation of discourse obligation for the listener to respond)
- *Cancel*: closes the current DU as ungrounded

Figure 3.3.: Grounding acts in Traum's finite state-based grounding model [Tra94]

3.1.2. The finite state-based grounding model of Traum

Traum [TA92, Tra94, Tra96, Tra99] developed one of the first implementable grounding models within his work for the project of TRAINS-93, which was briefly described in the section 2.1.3. His model addresses the first problem of Clark's model (recursive Acceptance), and is implemented for a conversational agent in a collaborative planning system.

By analyzing the TRAINS corpus Traum found out that only certain patterns of utterance sequences are possible in dialog, e.g., a speaker can not acknowledge her own immediately prior utterance and the listener's request for the speaker to repair her own utterance usually creates an obligation for the speaker to actually do so. This behavior suggests that, for the grounding state of a certain dialog segment, the number of possible state transitions is finite. Motivated by this finding, Traum proposes to replace the structure of Contribution with *discourse unit* (DU). DUs are the units of dialog at which grounding takes place and are composed of utterance-level actions rather than Presentation and Acceptance phases. These utterances perform grounding acts which change the grounding state of a DU. Grounding acts that can be performed are listed in Fig.3.3 and possible grounding states of a DU in Fig.3.4.

The grounding process follows the principle of a finite state-machine: Each incoming grounding act triggers a state transition in the DU. Once a DU arrives at its final state (*state F*) this unit of dialog is grounded. If the DU reaches the *state D*, then this unit of dialog is abandoned without being grounded. Any other state indicate that the DU needs one or more utterances performing certain grounding acts to arrive at the final state of grounding. This model can thus, in any state of the DU, precisely predict what grounding act is needed (or will follow in the ideal case). Figure 3.5 illustrates the state transitions of DUs.

Traum's model solves the problem of recursive Acceptance by allowing autonomous acknowl-

- *State S*: the DU is not yet initiated
- *State F*: the DU is grounded
- *State D*: the DU is abandoned although it is not yet grounded
- *State 1 - 4*: the DU needs one or more utterances to be grounded

Figure 3.4.: Grounding states of a Discourse Unit in Traum's finite state-based grounding model [Tra94]

State	Entering Act	Preferred Existing Act
S	———	Initiate^{I}
1	Initiate^{I}	Ack^{R}
2	ReqRepair^{R}	Repair^{I}
3	Repair^{R}	Ack^{I}
4	ReqRepair^{I}	Repair^{R}
F	Ack$^{\{I, R\}}$	Initiate$^{\{I, R\}}$ (next DU)
D	Cancel$^{\{I, R\}}$	Initiate$^{\{I, R\}}$ (next DU)

A

Next Act	In State						
	S	1	2	3	4	F	D
Initiate^{I}	1						
Continue^{I}		1			4		
Continue^{R}			2	3			
Repair^{I}		1	1	1	4	1	
Repair^{R}		3	2	3	3	3	
ReqRepair^{I}			4	4	4	4	
ReqRepair^{R}		2	2	2	2	2	
Ack^{I}				F	1*	F	
Ack^{R}		F	F*			F	
ReqAck^{I}		1				1	
ReqAck^{R}				3		3	
Cancel^{I}		D	D	D	D	D	

* repair request is ignored

B

Figure 3.5.: Discourse Unit transitions. (A) meaning of state units, (B) DU transition diagram (I = initiator, R = responder) [Tra94]

edgment acts (Acceptance) that do not require further Acceptance. With the finite state-machine, this model also enables a clear definition of grounding conditions for each given DU[1].

Nevertheless, Traum's model is also subject to limitations. Despite different representations, DUs are conceptually comparable to Clark's idea of *embedded* Contribution. This means, when-

[1] Note, this grounding model is a part of Traum's *multi-level theory of conversation acts* [TA92]. In this theory, different aspects of a dialog are represented using different types of *conversation acts*, e.g., grounding acts for grounding and turn-taking acts for coordination of turns. Domain tasks are not directly modeled using grounding acts and the finite state-machine, so that the domain task problem of Clark's model is not relevant in Traum's model.

ever there is a problem in understanding during the dialog, this unit of grounding is extended with repair utterances of dialog participants that also need to be understood, i.e., grounded, before the entire unit can be grounded in the end. This means that a DU can be in the state F several times. For example, in the transition diagram in Fig. 3.5, given the next act of *Ack* in the state F, the DU state remains F. These transitions make it difficult to determine whether a DU is definitely grounded or not since the state F can indicate both the groundedness of a conversational repair and of the entire DU. This situation would require a mechanism similar to a push-down automaton so that a repair can be pushed and popped before the entire DU is grounded.

3.1.3. The exchange model of Cahn and Brennan

Cahn and Brennan [Cah92, BH95, CB99] adapt Clark's model to task-oriented dialog in HCI and addresses his second problem of unclear relationship between the meta-process of grounding and the domain. Their solution involves two strategies: (1) specification of parameters that determine the grounding criteria, and (2) the augmentation of the contribution model with a task structure *Exchange*.

As to the first strategy, Cahn and Brennan propose that a speaker considers the following two issues when evaluating the conversational evidence of understanding that was provided by the listener in her reply :

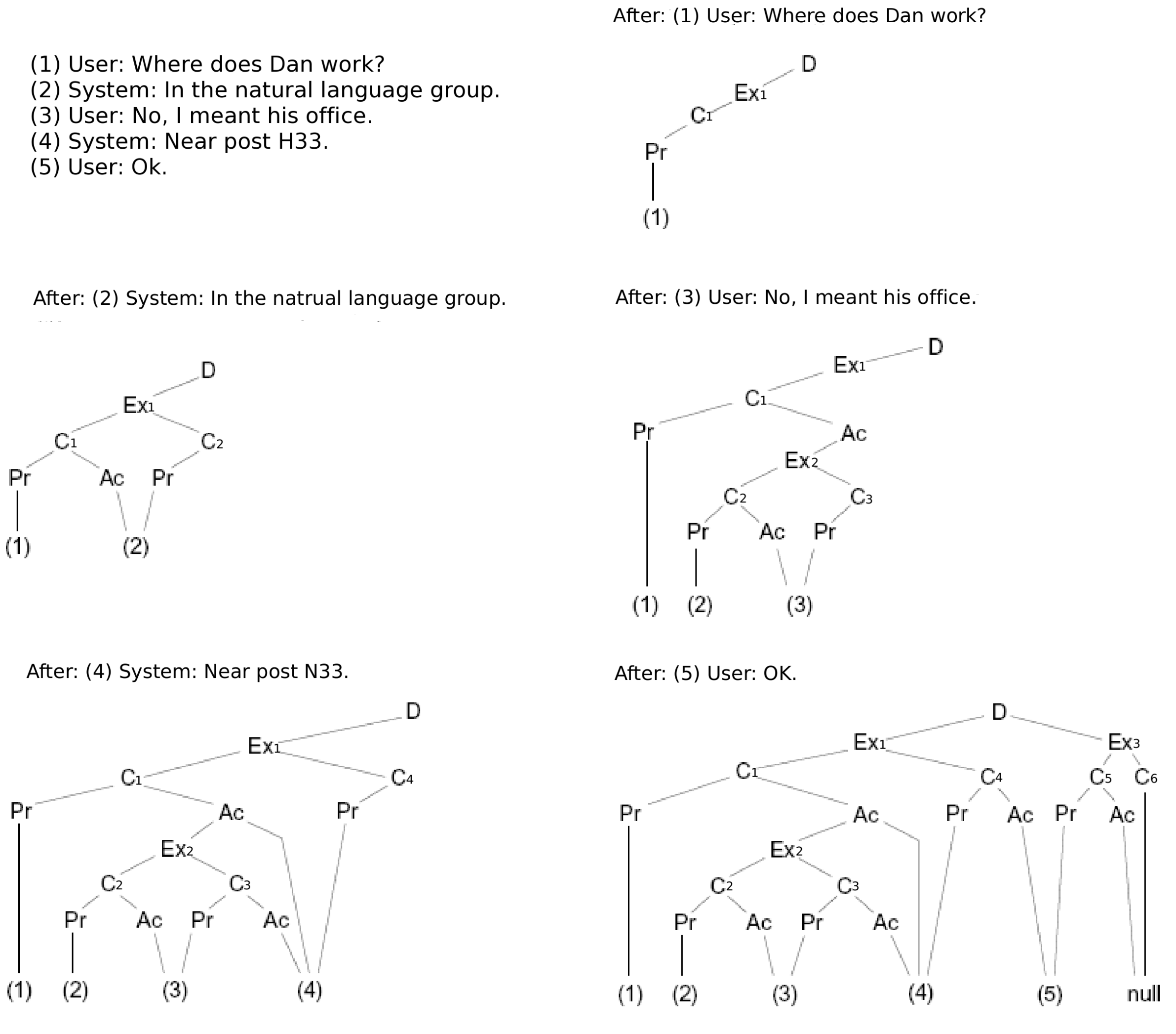
- whether the listener understood the speaker's Presentation;
- whether the reply of the listener is *conditionally relevant* to the speaker, i.e., whether the speaker accepts it as a relevant domain and conversational move.

The evaluation results can be either (1) the Presentation was understood and the reply is conditionally relevant or (2) the Presentation was understood but the reply is not conditionally relevant or (3) the Presentation is not understood. Only in case (1) the speaker's grounding criteria are considered as fulfilled. This means, only if the speaker thinks that her Presentation was understood and the reply was conditionally relevant, she is sure that her Presentation is grounded by the listener.

For the second strategy, Cahn and Brennan augment the contribution model of Clark with Exchanges. This concept is motivated by the observation that Contributions appear in pairs most of the time and the second Contribution does not only serve as evidence carrier for understanding but also as task executor that finishes the task initiated by the first Contribution. In the Exchange model, each Exchange is a pair of Contributions that are initiated by different dialog participants. The first Contribution initiates or defines a task and the second one completes or executes a task[2]. These tasks can be domain or conversational tasks (e.g., repair) and correspond to the *discourse*

[2] This structure of exchange is motivated by *adjacency pairs* proposed by Schegloff and Sacks [SS73] who state that utterances tend to occur in meaningful pairs that, together, accomplish a single collaborative task.

segment purpose in Grosz and Sidner's *focus space model* [GS86]. The tasks thus represent the idea that two Contributions contribute to a common purpose and are, therefore, structurally linked with each other. This discourse model can be portrayed as a graph (see Fig. 3.6), which is similar to Clark's graph. Here, two Contributions are linked at the root to represent an Exchange and the leaf nodes that are connected to utterances represent, as in Clark's model, the progression of understanding. The operations in this graph are determined by two variables: (1) whether the previous utterance is grounded,or not (according to the grounding criteria discussed above) and (2) whether the incoming utterance defines or executes a task. Based on the value of these two variables four operations can be carried out for one incoming utterance U_n (Fig. 3.7).


(1) User: Where does Dan work?
(2) System: In the natural language group.
(3) User: No, I meant his office.
(4) System: Near post H33.
(5) User: Ok.

After: (1) User: Where does Dan work?

After: (2) System: In the natrual language group.

After: (3) User: No, I meant his office.

After: (4) System: Near post N33.

After: (5) User: OK.

Figure 3.6.: The private model of the system concerning a clarification of task definition in the exchange model [Cah92] (D = dialog, Ex = Exchange, C = Contribution, Pr = Presentation and Ac = Acceptance).

Cahn and Brennan emphasize that this discourse model only represents the private model of *one*

- *create an Exchange* : create a new Exchange with U_n as the first Presentation [a]
- *close an Exchange* : complete an Exchange with U_n as the second Acceptance
- *embed an Exchange* : embed an Exchange into another one to carry out repair (with U_n playing various roles)
- *unlink an utterance*: re-allocate U_{n-1} (the previous utterance) that was previously linked with an Exchange

[a] Since each Exchange is composed of two Contributions in the original sense of Clark, each Exchange also includes two Presentations and two Acceptances, respectively

Figure 3.7.: Operations that can be carried out on the the discourse graph

dialog participant and is revisable during a dialog. Figure 3.6 reveals this point and presents the private model that the system builds and revises step by step during the dialog with a user. Here, the user first initiates Exchange Ex_1 by defining a task (a question) with Contribution C_1 and the user utterance is viewed as the Presentation of C_1. In (2), the system assumes that it correctly understands the user's Presentation and the system's answer also successfully executes the task that the user defined in C_1. This assumption is represented in the discourse structure as the link between the utterance (2) and the Acceptance of the C_1 *and* as the Presentation of C_2, which is intended to be the task executing Contribution of Ex_1. However, the user contradicts the system's answer in (3) so that the system revises its structure. Now C_2 is no longer the task executing Contribution of Ex_1, but the task defining Contribution in the new Exchange Ex_2 which is initiated to resolve the misunderstanding as a conversational task.

The exchange model clearly defines the relationship between the progression of understanding and the task execution, which is crucial for modeling task oriented dialog for computer applications. However, this model takes over the concept of recursive Acceptance as Clark proposes and thus inherits its problem that the grounding process does not end in a reasonable way. To overcome this problem, Cahn and Brennan introduce artifical structures as the end of the dialog. Taking the example in Fig. 3.6, in the last step, the user's utterance "Ok." is the Acceptance of C_4, which signals the user's understanding of system's Presentation and satisfaction with its task execution result. However, based on the recursive principle of Acceptance, this user utterance has to play the role of Presentation for a new Contribution, in this case C_5, which should be the task defining Contribution of Ex_3. The problem is, neither C_5 nor Ex_3 actually exist because the dialog already arrives at its end. C_5 and Ex_3 are thus artificial structures that do not match the dialog progress. Besides, it is not intuitive to consider the utterance "Ok" as having the function of defining a task. Although Cahn and Brennan claim that this model can be represented as a stack, too, it is considerablely difficult to actually do so because of the recursive nature of

Presentation and Acceptance. Another severe problem with this model is its inability to handle multiple utterances in one turn. With the recursive structure of the model, it is not clear what happens, when a dialog participant can accept only a *part* of her dialog partner's Presentation but has to initiate a new Exchange to clarify another part of the Presentation.

In fact, the Contributions do not have any specific functions in the exchange model at all: Presentation and Acceptance are responsible for ensuring understanding and Exchanges manage task execution. The existence of Contributions only seems to be justified by its meta function of connecting Presentation and Acceptance. But do they have to be connected by an extra structure of Contribution? Is it possible to model the pair of Presentation and Acceptance as being in a mini-state-machine, which represents a small grounding unit and possesses one start state triggered by Presentation and one final state triggered by Acceptance? If yes, this would mean to combine the concepts of Traum's finite state-model and the exchange model of Cahn and Brennan. Would this combination bring any advantages over the existing models? The following section examines this possibility and proposes a new grounding mechanism, which is further extended with the ability to handle multi-modality.

3.2. The MMPDA model

This section describes a new computational model of grounding, the MMPDA model, that attempts to combine the finite state-model of Traum and the exchange model of Cahn and Brennan. Furthermore, this model is augmented with a new structure that accounts for the multi-modality of dialog. In section 3.2.1 the key notions of this model are introduced and in section 3.2.2 the issue of multi-modality is discussed. Finally, in section 3.2.3 the whole picture of the model is depicted using the notions discussed in the two preceding sections.

Below, in abstract discussions concerning a segment of a dialog, the term "initiator" refers to the dialog participant who initiates an account and awaits it to be grounded while "responder" is her dialog partner who is expected to show evidence of understanding. The word "contribution" is used in its original sense of the general act of contributing instead of in the sense of Clark's contribution model.

3.2.1. Key notions

This subsection addresses four key notions of the new grounding model: *grounding unit*, *grounding relations*, *grounding criteria* and *types of Acceptance*. The notion of grounding relations is the central novel construct of the MMPDA model and will be discussed in detail. The other three notions have already been used in existing grounding models and the discussion here mainly aims at specifying their meanings in the MMPDA model. All of the four notions, however, serve as the basis for the MMPDA model and will be referred to again in section 3.2.3 as the whole picture of the model is depicted.

Grounding unit

The grounding unit, i.e., the unit of dialog at which the grounding takes place, is the Exchange. In contrast to Cahn and Brennan's model, the MMPDA model abandons the structure of Contribution and each Exchange in this model consists of two *dialog act level actions*, e.g., an instruction "Please close the door." and the following confirmation "OK.", or a question "What time is it?" and the subsequent answer "Ten o'clock." Note, a dialog act is "smaller" than a turn because a dialog participant can produce several dialog acts in one turn, e.g. "It is really a nice weather. How about a walk in the park?" This turn is viewed as containing two dialog acts: a statement and a question.

The two dialog acts contained in an Exchange are created by different dialog participants. From the view of grounding, a initiator creates the first act, which *initiates* an Exchange and plays the role of Presentation, and the responder creates the second one, which *grounds* the Exchange and plays the role of Acceptance. From the view of domain tasks, the Presentation initiates a task and the Acceptance executes the task. As in Cahn and Brennan's model, these tasks can be either conversational tasks (such as repair) or domain tasks. Each Exchange can be in one of the following two states:

- *Not grounded*: the Exchange is initiated, but not yet grounded. An Exchange is in this state when its presenting dialog act (Presentation) is available, but not its accepting dialog act (Acceptance);
- *Grounded*: the Exchange is grounded. An Exchange is in this state when both Presentation and Acceptance are available.

According to the above specifications, the utterance "It is really a nice weather. How about a walk in the park?" initiates two Exchanges each of which needs to be grounded, as illustrated in Fig 3.8.

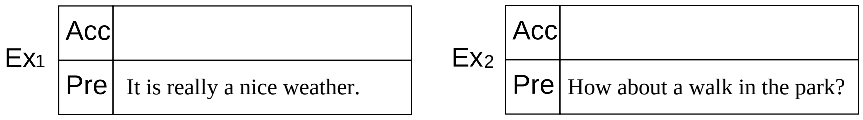


Figure 3.8.: Exchanges (Ex: Exchange, Pre: Presentation, Acc: Acceptance)

Although similar terms as in the existing grounding models are used in this new model, they have slightly different meanings than their ancestors. Firstly, the grounding unit of Exchange is "smaller" than Traum's Discourse Unit because it always contains only two elements. Secondly, an Exchange represents a task that can not be divided into subtasks any more. This means, repair effort of dialog participants such as clarifications are *not* part of an existing Exchange, instead, they initiate a new Exchange. In short, the grounding unit of Exchange is a quite *local* construct in terms of both grounding and task. The question of how they are connected during a dialog is answered below.

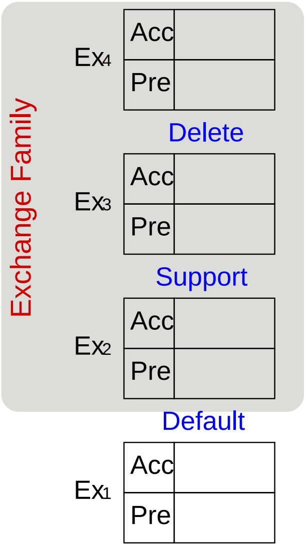


Figure 3.9.: Connecting Exchanges: Ex_{n+1} is the Mother Exchange of Ex_n and the Exchanges against the grey background belong to an Exchange Family.

Grounding relation

People participate in a dialog by performing operations with Exchanges which, apart from the first Exchange in the dialog, are usually in certain relation to previous Exchanges. Some Exchanges are initiated to clarify grounding failure that arose in previous dialog segments while some other Exchanges aim at canceling the dialog partner's grounding effort (e.g., by saying "forget it, it is not important."). Analysis of dialog examples suggests that the number of such relations is finite and four such relations are most frequent: *Default*, *Support*, *Correct* and *Delete*. These relations indicate *whether and how a given Exchange assists previous Exchanges with their grounding* and are called *grounding relations*.

Before the detailed discussion about the four grounding relations, a few terms need to be clarified first: Each Exchange has a *Mother Exchange*, which is the Exchange that is the top Exchange before the current Exchange. In this relation, the current Exchange is called *Son Exchange*. Grounding relation is the relationship between a Mother Exchange and its Son Exchange. Besides, each Exchange can belong to an *Exchange Family*, which is a group of at least three Exchanges that are connected via non-Default grounding relations. In Fig. 3.9, the Exchanges are named according to their creation time and, thus, Exchange Ex_2 is the Mother Exchange of Ex_3, Ex_3 is the Mother Exchange of Ex_4, and so on. Besides, Ex_2, Ex_3 and Ex_4 construct an Exchange Family because they are connected via non-Default grounding relations. When an Exchange is grounded, some actions can be carried out on its Mother Exchange and / or on its Exchange Family. Whether to carry out these actions and what actions should be selected depend on the grounding relations. In the discussion below, if Ex_n has grounding relation r to its Mother Exchange, then it is called *r Exchange* , e.g., Default Exchange is an Exchange that has Default relation to its Mother Exchange.

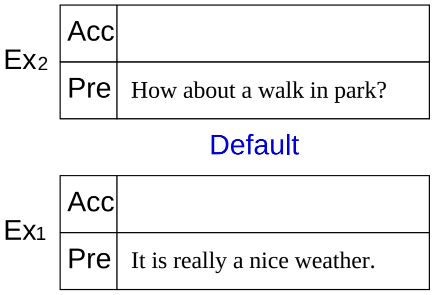


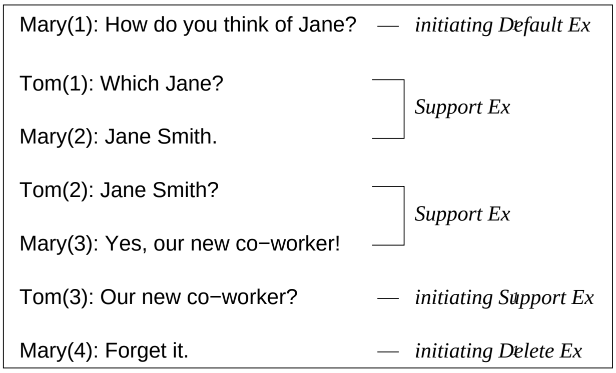
Figure 3.10.: Exchanges with grounding relation Default.

Default: Some Exchanges are initiated without particular function of assisting its Mother Exchange with its grounding, e.g., Exchanges that introduce new topics into the dialog. In Fig. 3.10, although the Presentation in Ex_2 seems to be motivated by the Presentation of Ex_1 in the pragmatical sense, it does not help the understanding of Ex_1. Ex_2 thus has grounding relation Default to its Mother Exchange Ex_1. If a Default Exchange is grounded, no further operations need to be carried out on its Mother Exchange.

Support: A Support Exchange is initiated to support the grounding process of its Mother Exchange by providing further information on the issue addressed in the Mother Exchange. A typical example of such an Exchange is one motivated by a clarification question like “I beg your pardon?”, which aims at collecting precise information on what was said in the Mother Exchange. If a Support Exchange is grounded, the dialog participant who was supposed to ground the Mother Exchange will retry the grounding process of the Mother Exchange with the information that is collected in the Support Exchange.

Correct: Sometimes dialog participants can erroneously believe that they successfully grounded an Exchange or they already provided correct information for the other dialog participant to ground although it is not true. In such situations they initiate Correct Exchanges to correct the Mother Exchange, e.g., in case of third-turn repair, the initiator of the initial Mother Exchange realizes that her Presentation was misunderstood and then initiates a Correct Exchange by saying “No, I meant...”. If a Correct Exchange is grounded, the dialog participant who was supposed to provide Acceptance for the Mother Exchange will retry the grounding process of the Mother Exchange with the information that is collected in the Correct Exchange.

Delete: During a dialog it can occur that dialog participants want to cancel their joint grounding effort for some reasons. They signal this intention by initiating Delete Exchanges. Such Exchanges are unique in the sense that they do not only affect their Mother Exchange once they are grounded: they also have the power to cancel the need for grounding for all members of its Exchange Family. This policy is motivated by the observation that cancellation of grounding effort often occurs due to frustration of dialog participants when no common ground can be established in spite of repeated attempts (repeated Support or Correct Exchanges). For example, in Fig 3.11, Tom can not understand who is Jane Smith although Mary has provided more and more information before she gives up. In the language of the MMPDA model this is to say, Tom can



Mary(1): How do you think of Jane? — *initiating Default Ex*

Tom(1): Which Jane?

Mary(2): Jane Smith.

Tom(2): Jane Smith?

Mary(3): Yes, our new co−worker!

Tom(3): Our new co−worker? — *initiating Support Ex*

Mary(4): Forget it. — *initiating Delete Ex*

Figure 3.11.: Cancellation of grounding effort with a Delete Exchange

Relation	Purpose for Grounding	Purpose for Tasks	Post-operation after grounding
Default	no assisting function for the grounding of the Mother Exchange	initiate a task	wait for next utterance
Support	facilitate the grounding of Mother Exchange by providing more information	contribute to the task execution	retry to ground the Mother Exchange.
Correct	facilitate the grounding of Mother Exchange by correcting it.	contribute to the task execution	retry to ground the Mother Exchange.
Delete	cancel the joint grounding effort for members of the Exchange Family.	delete a task	give up grounding of the Exchange Family.

Table 3.1.: Grounding relations

not ground Mary(1) although he has initiated three Support Exchanges to assist this grounding process. In Mary(4), Mary initiated a Delete Exchange which not only attempts to delete its immediate Mother Exchange, but also all the other Support Exchanges until the initial Default Exchange in which Mary introduced the question into the dialog, Mary(1). As can be seen, if a Delete Exchange is grounded, no effort will be made to ground members of its Exchange Family.

Note, Grounding relations are established when an Exchange is initiated which means that it is determined *before* an Acceptance is available. Therefore, to some degree, they can be viewed as an attribute of Presentation and, thus, only affects the initiation of a task: A Default Exchange initiates a domain task, Support and Correct Exchanges initiate (often) conversational tasks (contributing to the execution of the domain task) and Delete Exchanges attempt to cancel the task.Whether these tasks can be executed depends on the availability of the corresponding Acceptance. (For a summary of the grounding relations, see Table 3.1).

As stated earlier in this section, grounding relations describe whether and how a given Exchange assists the grounding process of its Mother Exchange. Two important issues have been left out in the general description here, namely, *the issue of Exchange initiator*, i.e., who of the dialog participants initiates the Exchange, and *the issue of timing*, i.e., whether an Exchange is created *before* or *after* the responder's reply. Taking into account these two issues the grounding relations proposed here are able to model a large number of repair behaviors in dialog, as will be shown in the section 3.3.2. For now, it is more important to look at how dialog participants make their decisions as to whether an Exchange is grounded or not because these decisions affect the choice of the grounding relation of the next Exchange.

Grounding criteria

The determination of how various factors affect dialog participants' choice of their grounding criteria is not trivial. As Traum points out [Tra99], there are two central questions: how much grounding is enough for the current purpose and how important is it that the grounding reaches that level. Although different scales have been proposed [DTS96, ANA92], it is still a challenging question. For the purpose of the current work, this problem is side-stepped and a simple set of criteria, similar as proposed by Cahn and Brennan, is adopted.

As Brennan [BH95] suggests, "understanding" (which means grounding) occurs on different levels of communication. She identifies seven states for a task-oriented spoken dialog system: not attending, attending, hearing, parsing, interpreting, intending, acting and reporting. These states are proposed to imitate cognitive processes that are involved when a human responder generates a reply. Similarly, in the MMPDA model, for the purpose of establishing parameters that determine grounding criteria, three broad categories of cognitive activities are identified according to their different requirements on knowledge:

- *language understanding*: This category includes cognitive activities that involve linguistic interpretation of what the initiator said, which is roughly equivalent to parsing and interpreting in Brennan's categorization. The knowledge required for this analysis is mainly linguistic knowledge and the analysis can succeed or fail.

- *concept understanding*: If the linguistic interpretation succeeds, it has to be further mapped to appropriate cognitive concepts of the responder in the relevant domain, e.g., mapping the word "cup" to the concept of a bowl-shaped drinking vessel. Brennan's state of "intending" is a similar term. This mapping requires a general model of the domain and can be successful or unsuccessful.

- *decision making*: Making the decision as to what a reply should be generated demands detailed domain knowledge on what is possible *right now* and is affected by emotion and situation. As proposed by Cahn [Cah92], the decision of a reply can be either conditionally relevant or irrelevant (compare to section 3.1.3 on page 37).

This categorization only serves as orientation and there may be intersections between them. Nevertheless, it provides a handy criteria catalog for grounding: *If the language of the initiator can be successfully understood, the appropriate cognitive concept can be successfully identified* *and a conditionally relevant reply can be generated, then an Exchange can be grounded by the responder*. Otherwise, it can be predicted that the responder will initiate a new Exchange with appropriate grounding relations to assist or cancel this grounding process. In comparison to Cahn and Brennan's criteria, the issue of cognitive concept is added here because of its practical relevance for learning scenarios of a robot, as will be discussed in chapter 4. These criteria are only guidelines for the determination of groundedness and their interpretation is subject to domain and applications, e.g., whether the language which should be understood includes body language, when is a reply conditionally relevant, and so on.

It is important to keep in mind that these criteria only address how the *responder* determines whether she can ground the Exchange initiated by the initiator. These criteria can not predict whether the initiator really views the responder's reply as a qualified Acceptance. The reason for this discrepancy is the (possibly) different grounding criteria that are hold by the initiator and the responder. This difference can result from their different domain knowledge, personal preferences and so on. Thus, the whole picture of how an Exchange is grounded involves that the responder feels being competent in generating a reply that meets her personal grounding criteria *and* the initiator also concludes that it actually also meets her grounding criteria.The groundedness of a proposition is thus subject to assessment of *both* dialog participants and when either of them has a problem, it can be predicted, she will initiate a new Exchange of appropriate grounding relations.

Types of Acceptance

As discussed in section 3.1.1, Clark identifies five types of Acceptance and they are of different strengths (page 32). As discussed in last section, this categorization is problematic especially in combination with his strength of evidence principle (also see [Tra96] and [TD98]). To solve this problem, the classification of Acceptance types in the MMPDA model is done on the meta level: classification based on *the target of the responder's reply*. Analysis of dialog examples reveals that, if the Presentation of a given Exchange Ex_n contains the proposition P_n, the Acceptance of Ex_n can address P_n itself, a new proposition, say P_{n+1}, or nothing specific:

- P_n: The reply of the responder directly addresses P_n and conveys a certain domain relevant intention, e.g., if P_n is a question, answering it can convey the responder's intention to inform the initiator of her opinion; or if P_n is an instruction, confirming it expresses the responder's agreement with or commitment to the instruction. This is the most direct way to ground an Exchange and it is called *Type* P_n (see Fig. 3.12 (a));

- P_{n+1}: The reply of the responder introduces a new *domain* task into the dialog which initiates a new *Default* Exchange Ex_{n+1}[3]. The new Default Exchanges are roughly equivalent to Clark's "initiation of the relevant next turn" and have the special structural effect that

[3] Such Exchanges should not be confused with those that initiate non-Default Exchanges, e.g., clarification questions that initiate new Support Exchanges, are not viewed as Acceptance.

Ex$_n$ *is grounded although the position of its Acceptance is not yet occupied (see Fig. 3.12 (b))*. This type of Acceptance is called *Type* P_{n+1}. The identification of such an Acceptance is sometimes not straightforward because the choice of the grounding relation between Ex_n and Ex_{n+1} depends on the individual interpretation of the owner of the discourse model. For example, if the initiator of Ex_n expects to get an Acceptance of Type P_n while the responder proposes a new task in Ex_{n+1}, then for the initiator, the Ex_{n+1} can be interpreted as a Delete Exchange instead of a Default one.

- θ: The reply of the responder does not directly address anything and the existence of this reply is merely to keep the dialog going on (compare to Clark's Acceptance type 1, continued attention). This implicit way of grounding often involves behaviors whose propositional meaning can not be easily identified as to how it contributes to the task execution, e.g., continued attention. Such Acceptance is of *Type* θ (see Fig. 3.12 (c));

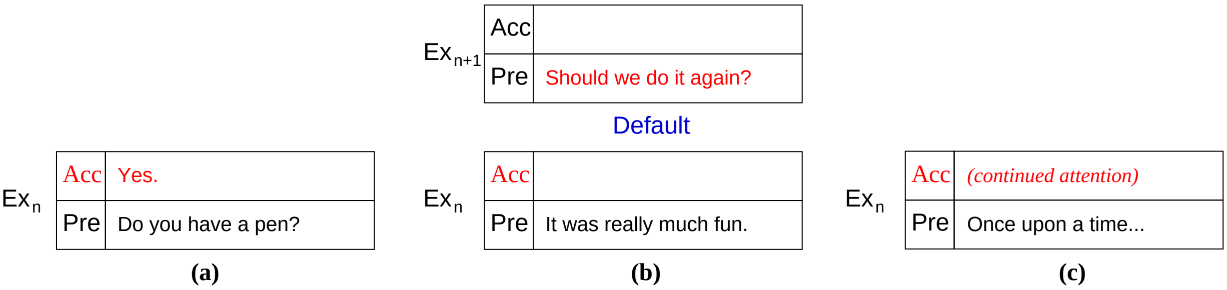


Figure 3.12.: Acceptance types: (a) Type P_n, (b) Type P_{n+1}, (c) Type θ (The red-lined part of each Exchange grounds the Exchange)

It is not a trivial task to determine which type of Acceptance is appropriate to use in a certain situation. However, it can be reasonable to assume that there are certain correlations between grounding relations and the types of Acceptance that are required. For example, the existence of a Support or Correct Exchange always indicates some grounding problems during the dialog which may require a more explicite type of Acceptance like type P$_n$ rather than type P$_{n+1}$. In case of a Delete Exchange, it is likely that the dialog participants ground it directly by proposing a new task since the Exchange initiator intends to delete the ungrounded current issue anyway. The MMPDA model simplifies these correlations with two assumptions:

1. The Acceptance of a Support or Correct Exchange is never of type P$_{n+1}$, and
2. A Default Exchange immediately after a Delete Exchange is viewed as the Acceptance of the Delete Exchange (type P$_{n+1}$).

This classification of Acceptance types has the advantage that it stays on the meta level of a dialog and avoids a general statement as to how they can be realized, e.g., in what linguistic structure or in which modalities (as in Clark's proposal). It is important to differentiate this because it enables a clear description of the causal relationship between the Acceptance types and their effects in the discourse model without over-simplifying dialog behaviors in general. For the purpose of implementation, it also allows more flexibility as to the realization of Acceptance.

So far, the most important notions of the new model have been discussed, but one is still left out: the issue of multi-modality. Empirical studies show that non-verbal behaviors play an active role in the grounding process. For example, Dillenbourg et al. [DTS96] found out that grounding is not bound inside one modality and information presented in a modality can be grounded by an act in another one. Nakano et al. [NRSC03] state that non-verbal behaviors are used as grounding cues and can sometimes fulfill the grounding function without involving verbal behaviors. In the MMPDA model so far, Presentation and Acceptance only represent dialog acts and do not seem to be able to account for such behaviors. The following section addresses this issue and proposes an extension of the concept.

3.2.2. The issue of multi-modality

This section first analyzes the existing multi-modality modeling approaches for embodied interaction and argues that it is beneficial to extend the concept of grounding to account for multi-modality. Then, two extensions are presented in detail: widening the specification of common ground and modeling the most basic unit of an embodied interaction with Interaction Unit.

Motivation

As mentioned in the section 2.2.2, Cassell [Cas00] tackles the issue of multi-modality in the context of embodied interaction by grouping information involved into categories and handle them separately. Cassell differentiates between interactional and propositional information. In her generic architecture for conversational virtual agents (page 19), these two types of information are processed by two different system modules. The first one is the *interactional processor* that mainly processes non-verbal contributions of the participants that change the “meta-state” of the interaction, e.g., who has the turn and whether the turn is suspended. The second module is the *propositional processor* and it analyses verbal contributions that mainly contribute to the propositional discourse. However, it is not always easy to make a clear distinction between information types and one single interaction contribution often has to be analyzed in terms of its relevance as several types. For example, the devision of tasks between interactional and propositional processors may fail to handle situations where the verbal and non-verbal contribution co-carry a propositional meaning, e.g., if the user says “Show me *that* room.” and points to it with a deictic gesture. To avoid this problem the system has to analyze each non-verbal contribution in terms of whether it changes the interactional state or it contributes to a proposition, which is not always easy.

Rather than addressing *different* functions of contributions in an embodied conversation, it can be beneficial to address their *common* function instead. And this is the *evocative function* of behaviors involved in an interaction. Most human behaviors involved in an embodied interaction, whether they contribute to the regulation of the communication itself or to the propositional discourse, have an evocative function. This means, *these behaviors place an obligation on other interaction partners to react* [All01, ANA92]. For example, if a responder Mary raises a hand indicating that she wants to speak, then the initiator Tom will usually stop speaking and release the turn. Here, the interactional information of raising a hand, as defined by Cassell, is generated by Mary and it has the function that Tom reacts to it by releasing the turn. Similarly, if Jane asks a question, her dialog partner Jack will usually reply to it or at least indicate his hearing. Here, Jane generates some propositional information which has the function that Jack addresses it properly. Whenever the evocative function of these behaviors can not be realized, e.g., in the above examples, if Tom does not release the turn or Jack does not answer the question, then the initiator of these behaviors will probably perform other behaviors so that these functions can be fulfilled. For example, in case of the over-active initiator Tom, Mary could generate some stronger contributions such as waving hands to attract more attention from him. In case of the silent responder Jack, Jane can initiate a question “Are you listening to me?” Dialog participants do this probably because they feel that their interactional or propositional information is not perceived or understood.

As can be seen, information that is exchanged between dialog participants in a multi-modal embodied interaction can contribute to either the regulation of the communication itself (interactional information) or to the propositional discourse (propositional information). In either case, the evocative function of the information can only be fulfilled when the information can be mutually perceived and understood. This similarity between interactional and propositional information exchange suggests that *a grounding model is able to account for both types of information exchange*. The advantage of doing so is that it is no more necessary to classify contributions of dialog participants into different types, which is often a challenging task as discussed above. A multi-modal grounding model thus would cover many aspects of embodied interaction and, in the meantime, provide a sophisticated discourse management mechanism. For this purpose, the MMPDA model is extended in two ways: (1) extending the definition of “common ground” and (2) modeling dialog contributions as Interaction Units.

Extension (I): common ground

Common ground originally refers to mutual knowledge and beliefs that dialog participants share based on what has been said during the dialog (see section 3.1). As Traum states, the issues of contact and attention are distinguishing features of embodied interaction. So the first that needs to be added in the original definition of common ground is the establishment of the physical possibility of communication, e.g., the (potential) dialog participants have visual or audio access to each another and both have the motivation to talk. At this stage (the contact layer in Traum’s model), what they mentally share is their awareness of the physical possibility of contact and their willingness to interact. The establishment of this type of common ground is often signaled by non-verbal interactional contributions of dialog participants, e.g., one walks into the vicinity of the other. After these pre-conditions are fulfilled, the communication channel is established. Then, during the interaction, what the participants share is their mutual understanding as to the interactional and propositional contributions of dialog participants, which can be multi-modal. As a summary, the definition of common ground is extended to:

> The common ground involved in an embodied interaction is what dialog participants mentally share and is exhibited by their multi-modal behaviors. It includes (1) their awareness of the physical possibility of contact and their willingness to interact and (2) their mutual understanding of the interactional and propositional information that is exchanged during the interaction.

The extended definition of common ground implies that dialog participants can not only provide Acceptance to what has been said propositionally, but also to other parts of the common ground. Taking the previous example, if Mary raises a hand, she generates a Presentation which is intended to establish the common ground with Tom that the turn is requested. If Tom releases the turn to Mary, he signals his Acceptance for Mary's Presentation and the common ground is established. However, if Tom does not release the turn, he does not provide Acceptance for the Presentation and Mary will probably initiate new Support Exchanges to help Tom ground her Presentation. This process is parallel to the case where Jane asks a question (creation of a Presentation) with the intention to establish the common ground concerning the question. If Jack replies in a satisfying way (Acceptance is available), the initiator Jane will take the common ground as established and no more effort needs to be made concerning this question. However, if Jack does not reply or his answer is not satisfying, Jane will initiate new Support Exchanges to facilitate Jack's grounding process.

Including the awareness of the physical possibility of communication into the definition of common ground means an extension of the grounding criteria which were specified in section 3.2.1 on page 45. This means, whether a dialog participant can provide Acceptance or not does not only depend on her ability to understand the language, to identify the cognitive concept, and to provide a conditionally relevant reply, but also depends on her ability to *perceive* the Presentation.

Extension (II): Interaction Unit

In previous discussions, the most basic units of a dialog have been identified as dialog act level actions and an Exchange is a pair of such actions. To account for multi-modality, the structure of dialog act is extended to model the cognitive process of language generation in a simplified way: as Interaction Units (IUs). In the following, the structure of such an IU and the process of generating it are discussed in detail.

The overall structure: An IU is a two-layered structure consisting of a *Motivation Layer* and a *Behavior Layer* (Fig. 3.13). On the Motivation Layer (MLayer), a motivation is conceived which

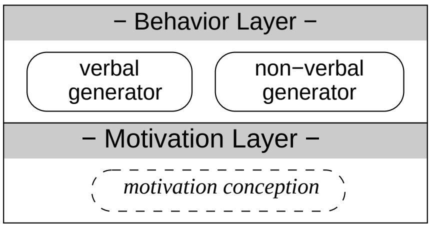


Figure 3.13.: The structure of an Interaction Unit (IU)

drives the generation of some behaviors on the Behavior Layer (BLayer). Note, a motivation can be *intentional* or *unintentional*. For example, if Mary looks sad and Tom asks her "Are you OK?", then Mary's sadness is also a dialog-related motivation. Of course, Mary may not intend to communicate her sadness originally, but the act of dialog between Mary and Tom is already established. Thus in this case, Mary's sadness is also viewed as a communication-related motivation. This is similar at the stage of creating a precondition for interaction: if Mary sits in the classroom and Tom walks into it so that Mary and Tom have visual access to each other, then the precondition of interaction is established, although Tom may not walk into the room on the purpose of interacting with Mary, originally[4]. During a dialog, dialog participants' intentional and unintentional motivations are manifested by behaviors that are generated on the BLayer.

The generators: A verbal and a non-verbal generator are located on the BLayer. They are responsible for generating spoken language and various non-verbal behaviors according to the motivation conceived, respectively. The two generators do not need to be instantiated at the same time, instead, it depends on the decision that is made on the BLayer as to what modality should be used to demonstrate the current motivation. This is to say, a dialog participant may express her motivations using one or more modalities. For example, if one smiles upon the Presentation of her dialog partner, her non-verbal generator on the BLayer of her IU is instantiated while the verbal generator is not. However, if she smiles and says something at the same time, then both generators on the BLayer are instantiated. Note, the relationship between the two generators represent the relationship between verbal and non-verbal conversational behaviors which is variable. Scherer and Wallbott [SW79] state that non-verbal behaviors can *substitute*, *amplify*, *contradict* and *modify* the meaning of a verbal message. Iverson et al. [ICLC99] has studied human gestures and identified three types of informational relationship between speech and gesture: reinforcement (gesture reinforces the message conveyed in speech, e.g., emphatic gesture), disambiguation (gesture serves as the precise referent of the speech, e.g., deictic gesture accompanying the utterance "this cup"), and adding-information (e.g., saying "The ball is so big." and shaping the size with hands). The specification of these relationships and the conditions for their existence are beyond the scope of this work. As will be shown in the next chapter, the focus of the implementation are the disambiguation (page 94) and the amplifying functions (page 105).

[4] Note, the fulfillment of the precondition of an interaction does not mean that the interaction is going to start in any case. In other words, one can only speak of "precondition for an interaction" if an interaction actually takes place.

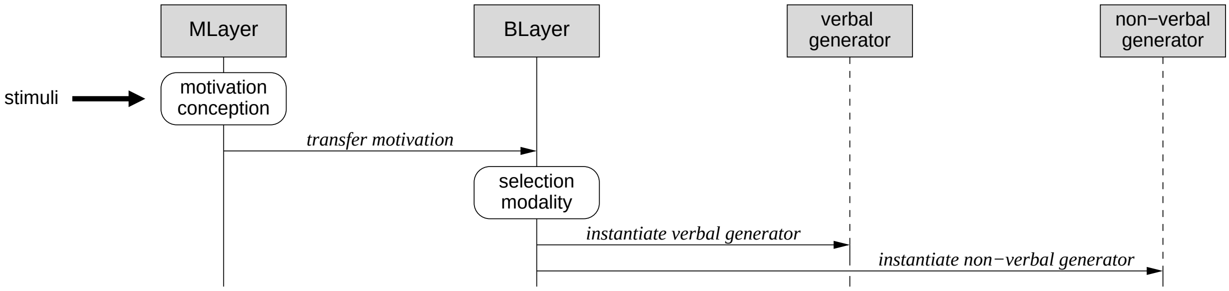


Figure 3.14.: Creating an IU based on self-motivation. Based on the modality selection result of BLayer either verbal, non-verbal or both generators can be instantiated.

The generation of IUs: During a dialog, a dialog participant either initiates an account or replies to accounts of her dialog partner. The contribution of a dialog participant can thus be *self-motivated* or *other-motivated*. In case of self-motivated contributions, a dialog participant creates an IU by conceiving a motivation on her MLayer. This motivation is then transfered to the BLayer and the BLayer decides which modality (or modalities) should be selected to demonstrate this motivation[5]. Based on this decision, either verbal, non-verbal or both generators are activated to construct a verbal message or a non-verbal signal or both. This process is illustrated as a UML sequence diagram in Fig. 3.14. In case of other-motivated contributions, the responder has to first analyze the IU that has been created by the initiator before generating her own IU. The goal is to understand the motivation of the initiator. In the language of IUs this is to say that the responder has to analyze the behaviors generated by the (verbal and/or non-verbal generators on the) BLayer of that IU to find out the content of the initiator's MLayer. The result of this attempt for understanding will be compared to the responder's personal grounding criteria[6]. This means, if the responder (1) succeeds in figuring out the content of the MLayer of the initiator's IU, (2) can associate this motivation with some of her cognitive concepts, and (3) feels competent in conceiving a conditionally relevant motivation as a reply, then the responder will provide Acceptance for the initiator's IU by generating a new IU. The responder does this in the same way as depicted in Fig. 3.14. If one of the three criteria can not be fulfilled, it can be predicted, the responder will create a new Presentation which initiates new Exchanges of appropriate grounding relations.

Recall the description of grounding units in section 3.2.1 (page 41), a grounding unit in the MMPDA model is an Exchange consisting of two dialog act level actions and a dialog act is a smaller unit than a turn. Similarly, as a multi-modal representation of a dialog act level action, an IU also only represents a "sub-turn" action. In the dialog example in Fig. 3.15, Tom's reply consists of two IUs: The first one is based on the motivation of informing Mary of his decision and is manifested by the non-verbal behavior of shaking head. The second one is based on his motivation of providing further information about the decision and this motivation is conveyed by an utterance.

[5] The determination of criteria for modality selection is a complex issue and is not addressed in the MMPDA model.
[6] given that the responder has already successfully perceived the initiator's contribution

Mary: Will you join us?
Tom: *(shakes head)* I have a meeting this afternoon.

Figure 3.15.: A dialog example

The structure of IU may appear trivial to some readers because it looks quite intuitive. In fact, even Clark already mentioned that evidence for grounding can be multi-modal. In the Mission Rehearsal Exercise project (page 21), Traum also uses visual and verbal evidence to perform grounding acts. But in few of existing discourse models, the multi-modality of interaction contributions is *explicitly represented*. As discussed in section 2.2.2 on page 18, researchers of the two strands of multi-modal dialog modeling focus on different aspects of such an dialog. For researchers who are interested in the modeling of relationship between individual interaction contributions, the representation of the discourse is often not that important, while for the researchers who are rather interested in the modeling of the entire discourse, the issue of modality realization is not essential, either. However, to realize human-like communication capabilities for virtual agents and robots, both aspects are crucial and should be studied in a unified manner. The structure of IU is proposed to bridge the gap between the strands with an explicit representation of the modality realization process in the discourse model. This structure has the potential to systematically represent modality selection and management processes during an interaction (compare to section 3.3.3).

Now, it is time to clarify how these quite local IUs are coupled into Exchanges, how the Exchanges are connected via grounding relations and how they change the discourse state during an interaction. The next section addresses these issues.

3.2.3. The whole picture

Two main issues are addressed in this section: *What is the overall structure of a discourse* and *How do IUs of dialog participants operate in this structure*. A short answer of the first question is that the discourse of an on-going dialog is represented as a stack with Exchanges (consisting of IUs) as stack elements. The answer of the second question is that this stack operates based on the principle of a push-down automaton. In the discussion below, IUs that play the role of Presentation and Acceptance in an Exchange are directly called “Presentation” and “Acceptance” respectively, while “IU” itself will refer to the contributions of dialog participants in general.

A push-down automaton [7]

Dialog participants contribute to a dialog by generating IUs that initiate or ground an Exchange. These Exchanges can be organized in *a stack which represents issues of the on-going dialog that need to be grounded*. During a dialog one participant proposes an IU for the other to consider and this process can be modeled as her creating an IU and pushing a corresponding ungrounded Exchange onto the stack. When the dialog partner provides a qualified Acceptance for this Exchange, both dialog participants can assume that the common ground concerning this Exchange is established and they do not need to consider it anymore. This process is roughly equivalent to popping this Exchange from the stack. So far, it looks quite plausible to model the structure of the discourse as a stack.

Two things have to be clarified concerning this structure: Firstly, as Cahn and Brennan emphasize, this structure only represents the view of *one* dialog participant, namely that of the *discourse holder*. All the operations on the stack are thus carried out by her based on her own grounding criteria. Secondly, the overall structure is grounded when the first Exchange that is pushed onto the stack is grounded, i.e., when the stack is empty. Since an Exchange consists of IUs, which represent fairly small meaning units, the stack can be emptied several times even during a short dialog.

As to the question of how such a stack operates, it can be decomposed into the following four sub-questions:

- What is the stack alphabet, i.e., what symbols can be pushed onto the stack?
- What is the input alphabet, i.e., what input signals can dialog participants generate?
- What states can the stack be in?
- What are the possible state transitions in this stack?

In the discussion below, the set of stack alphabet is denoted as Φ, the set of input alphabet as Σ, the set of states as Q and the set of state transitions is denoted as δ.

Stack alphabet: A stack contains *ungrounded* Exchanges. The most important attribute of an Exchange is its grounding relation because it justifies the existence of an Exchange. Thus, the stack alphabet includes Exchanges with four different types of such relations and they are abbreviated to *DefaultEx*, *SupportEx*, *CorrectEx* and *DeleteEx* below. The stack can, of course, sometimes contain no Exchange at all, which indicates that nothing is left for grounding between the dialog participants. ϵ is used for this situation. The set of the stack alphabet is thus:

$\Phi = \{\text{DefaultEx, SupportEx, CorrectEx, DeleteEx}\}$[8]

[7] This automaton is an optimized version to that proposed in [LWS06].

[8] ϵ is not included in Φ because it is a part of the definition of state transition, see the formal specification of state transitions on page 56

Input alphabet: What kind of IUs can be generated by the dialog participant during a dialog? First of all, they can generate Acceptance for an Exchange, denoted as Ex_n. As discussed in section 3.2.1 on page 46, there can be three types of Acceptance. In contrast to Type P_n and θ, which occupy the position of the Acceptance of Ex_n, type P_{n+1} Acceptance results in a new Default Exchange, say Ex_{n+1}, and thus directly modifies the structure of the stack. Therefore, it makes sense to distinguish between these two cases and view them as two different input types: $Acc_{n,\theta}$ and Acc_{n+1}. Additionally to input type of Acceptance, dialog participants can also generate Presentation, which initiates a new Exchange. If the new Exchange is Ex_{n+1} and it has grounding relation Default, then such an input is abbreviated as *DefaultPre*. Dialog participants can equally generate a Presentation to complement information contained in Ex_n, to correct it or to delete the joint effort to ground Ex_n. This means that also *SupportPre*, *CorrectPre* and *DeletePre* are possible input signals. Furthermore, dialog participants can also generate IUs that can not be categorized into one of the above categories. This is especially the case if the dialog partner of the discourse holder tries to provide Acceptance, but the discourse holder views it as being unqualified. Such an input signal is called *Unqualified*. As a summary, the set of stack alphabet is:

$$\Sigma = \{\text{DefaultPre, SupportPre, CorrectPre, DeletePre, Acc}_{n,\theta}\text{, Acc}_{n+1}\text{, Unqualified}\}$$

Note, if an Exchange should be pushed given an input signal of type Presentation, then a causal relationship exists between the type of the Presentation and the type of the Exchange that should be pushed: if *r*Pre is created, then *r*Ex will be pushed, e.g., if a DefaultPre is created, then a DefaultEx should be pushed.

States: To determine the states of the stack, it is necessary to first obtain a rough idea of how it works. A stack can contain some elements of the stack alphabet or be empty. The state of being empty is called state *E*. When the discourse holder contributes a initial DefaultPre into the dialog, a DefaultEx Ex_n is pushed onto the stack and her dialog partner is expected to provide Acceptance for this top Exchange. Now the stack is in the state of awaiting an Acc which is the state AA (awaiting Acceptance). If the dialog partner creates an $\mathrm{Acc}_{n,\theta}$, which is an input that satisfies the discourse holder's grounding criteria (see page 45), then Ex_n is popped and the stack returns to the state E. However, if the dialog partner is not able to create a qualified Acceptance she creates a SupportPre to facilitate the grounding process for Ex_n. Thereupon, a SupportEx Ex_{n+1} is pushed onto the stack and the discourse holder is now expected to provide Acceptance. Here, the stack is awaiting multiple Acceptance: one for Ex_{n+1} and one for Ex_n. This state is called AMA (awaiting multiple Acceptance). If the discourse holder successfully grounds the current top Exchange Ex_{n+1}, then this Exchange is popped from the stack which then returns to the state AA. Now the dialog partner tries to ground the Ex_n with the freshly collected information through Ex_{n+1}. If she succeeds, she pops it so that the stack returns to the E state again. As can be seen, the discourse stack can be in one of the three states: E, AA and

AMA.[9] When it is in state E, there are no symbols on the stack which means that nothing is left for grounding between the dialog participants. It is reasonable to assume that dialog participants strive to arrive at this state during a dialog. When the stack is in state AA it awaits the Acceptance for the only Exchange on the stack, and after this one is grounded the stack will return to the state E. When the stack is in state AMA, it not only awaits Acceptance for the top Exchange, but also Acceptance for Exchanges below. Depending on how many Exchanges are left on the stack, its following state can be both AA and AMA. As a summary, the set of stack states is:

$$Q = \{\text{E, AA, AMA}\}$$

state transitions: State transitions specify the next stack state given an input signal, a stack symbol and the current state. According to conventions concerning push-down automaton definition [Sip97], such transitions should be in form of:

$$(Q \times (\Sigma \cup \{\epsilon\}) \times \Phi) \longrightarrow (Q \times \Phi\ {*})$$

In the following discussion, the Exchanges involved are denoted as:

- Ex_{top}: the current top Exchange of the stack;
- $\text{Ex}_{topfamily}$: the Exchange Family of Ex_{top};
- Ex_{top+1}: the Exchange that should be pushed onto the stack;
- Ex_{top-1}: the Exchange on the stack that is below Ex_{top}.

Note, in some notations of push-down automaton transitions, a pop operation results in a ϵ being shown as the resulting top element on the stack. For example, transition $\delta(\text{q}_0, \text{b, A}) \longrightarrow (\text{q}_1, \epsilon)$ means: Given the state of q_0 and the top element of A, when the input signal is b, then the stack will transit to state q_1 and A will be popped. In the following, however, pop operations will be denoted *using the stack element that becomes the top element of the stack after the pop operation*. For example, $\delta(\text{q}_0, \text{b, A}) \longrightarrow (\text{q}_1, \text{B})$ means that given the state of q_0 and the top element of A, when the input signal is b, then the stack will transit to state q_1, A will be popped and the top element on the stack after this operation is B, which was the element right below A on the stack. In the specification of transitions below, B is represented as Ex_{top-1} in general. This notation is clearer especially when being used to describe dialog in practice (see hand-modeled dialog excerpts on page 66 and page 67).

[9] It makes sense to differentiate between the state AA and AMA because it reflects the difference between individual Exchanges and Exchange Families. If there is only one Exchange left on the stack, the stack state will always be AA. However, if it is an Exchange Family left, the state can be both AA and AMA, which depends on the next Exchange to be pushed onto the stack.

1	δ(E, DefaultPre, ϵ) $\longrightarrow$ (AA, DefaultEx)
2 (a)	δ(E, SupportPre, ϵ) $\longrightarrow$ (E, ϵ)
(b)	δ(E, CorrectPre, ϵ) $\longrightarrow$ (E, ϵ)
(c)	δ(E, DeletePre, ϵ) $\longrightarrow$ (E, ϵ)
(d)	δ(E, Acc$_{n,\theta}$, ϵ) $\longrightarrow$ (E, ϵ)
(e)	δ(E, AA, Acc$_{n+1}$, ϵ) $\longrightarrow$ (E, ϵ)

Table 3.2.: State transitions 1 and 2

Before one of the dialog participants generates an IU at the beginning of the dialog the state of the stack is E, meaning that there is nothing to be grounded. In this state, only a DefaultPre can be possibly generated by dialog participants and thus only a DefaultEx can be pushed onto the stack. The reason for this exclusivity is that DefaultPre initiates a new topic which can exist on its own, while any other input types require at least one preceding Exchange on the stack. Once the DefaultEx is pushed, the stack transits to the state AA as it is now awaiting a single Acceptance (transition 1 in Table. 3.2). For the sake of completeness, the transition for other input types is also specified as transition 2.

In the state AA, the dialog participants can generate any types of IUs. For example, if the first DefaultEx is pushed onto the stack by the stack holder, say Mary, her dialog partner, say Tom, can generate a SupportPre in case he can not understand Mary's DefaultEx and a second Exchange SupportEx is then pushed onto the stack (transition 3(b) in Table 3.3). This is the case of second-turn other-repair. It is also possible that Tom generates an Acc$_{n,\theta}$ for the existing DefaultEx so that it is popped and the state of the stack returns to E (transition 4(a) in Table 3.3)[10]. In case that Tom generates an Acc$_{n+1}$, which means that Tom accepts Mary's DefaultEx by addressing a new topic, a new DefaultEx is pushed onto the stack immediately after the Mary's initial DefaultEx is popped out of the stack (transition 5 in Table. 3.3). No transitions are specified here for the case when the Ex$_{top}$ is a SupportEx or a CorrectEx because of the first assumption that has been discussed in the context of Acceptance types in section 3.2.1 (page 47): *The Acceptance of a Support or Correct Exchange is never of type P$_{n+1}$.*

Note, in state AA, the initiator of the DefaultEx Mary can also generate other Exchanges before her first Exchange is grounded by Tom. This is the case if she contributes multiple IUs in one turn by pushing more than one DefaultEx or carries out first-turn repair by further pushing SupportEx or CorrectEx onto the stack. In this situation, Tom can address the multiple Exchanges either all at once or one after another so that transition 6 needs to be added (Table 3.4). Here, after a DefaultEx, SupportEx or CorrectEx is pushed, the state remains in AA. After these Exchanges are grounded, the state transits to E (see transition 4 in Fig. 3.3.)

In state AMA in which the stack is awaiting multiple Acceptance, either dialog participant can initiate further DefaultPre and thus push further DefaultEx onto the stack. The resulting stack

[10] For transition 4(b) and 4(c), compare to transition 6(b) and 6(c) on the following page

3 (a)	δ(AA, DefaultPre, DefaultEx) $\longrightarrow$ (AMA, DefaultEx)
(b)	δ(AA, SupportPre, DefaultEx) $\longrightarrow$ (AMA, SupportEx)
(c)	δ(AA, CorrectPre, DefaultEx) $\longrightarrow$ (AMA, CorrectEx)
(d)	δ(AA, DeletePre, DefaultEx) $\longrightarrow$ (AMA, DeleteEx)
4 (a)	δ(AA, Acc$_{n,\theta}$, DefaultEx) $\longrightarrow$ (E, ϵ)
(b)	δ(AA, Acc$_{n,\theta}$, SupportEx) $\longrightarrow$ (E, ϵ)
(c)	δ(AA, Acc$_{n,\theta}$, CorrectEx) $\longrightarrow$ (E, ϵ)
5	δ(AA, Acc$_{n+1}$, DefaultEx) $\longrightarrow$ (AA, DefaultEx) pop(Ex$_{top}$) and push(Ex$_{top+1}$)

Table 3.3.: State transitions 3, 4 and 5

6 (a)	δ(AA, DefaultPre, DefaultEx) $\longrightarrow$ (AA, DefaultEx)
(b)	δ(AA, SupportPre, DefaultEx) $\longrightarrow$ (AA, SupportEx)
(c)	δ(AA, CorrectPre, DefaultEx) $\longrightarrow$ (AA, CorrectEx)

Table 3.4.: State transition 6

states depend largely on the current Ex$_{top}$ on the stack. In case it is another DefaultEx, the incoming DefaultEx will be simply pushed onto the stack (transition 7 in Table 3.5). If the current Ex$_{top}$ is a SupportEx or CorrectEx, it is helpful to adopt the first assumption of Acceptance types: *The Acceptance of a Support or Correct Exchange is never of type P_{n+1}* (page 47). This means that the incoming DefaultEx is definitively *not* the Acceptance for the top Support or Correct Exchange and, rather, it initiates a new topic just for its own reason. In a real dialog, this would mean, e.g., that one ignores the repair effort of her dialog partner and proposes something else into the dialog, instead. It is, therefore, reasonable to assume that such a DefaultPre essentially results in pushing a *DeleteEx* onto the stack (transition 8 in Table 3.5). The reason why the resulting state in transition 8 can be both AA and AMA lies in the nature of the grounding relation Delete. Recall that a grounded Delete Exchange will remove all the members of its Exchange Family. This means, if all the remaining Exchanges on the stack belong to the Exchange Family of Ex$_{top}$, then once the top DeleteEx is grounded, there will be no other Exchanges to be grounded at all (state AA). However, additionally to the Exchange Family, there can be other Exchanges, most possibly DefaultEx. This occurs when a dialog participant pushes more than one DefaultEx onto the stack but only one of them is to be deleted. In this case, the resulting state of the stack is AMA. The difference between these two cases is illustrated in Fig. 3.16.

If the current Ex$_{top}$ is a DeleteEx and a DefaultPre is to be pushed, the second assumption in section 3.2.1 is adopted: *A Default Exchange immediately after a Delete Exchange is viewed as the Acceptance of the Delete Exchange (type P_{n+1}).* Taking into account this assumption, the input of DefaultPre grounds the top DeleteEx and its Exchange Family is popped before a new

7	δ(AMA, DefaultPre, DefaultEx) $\longrightarrow$ (AMA, DefaultEx)
8 (a)	δ(AMA, DefaultPre, SupportEx) $\longrightarrow$ (AA, DeleteEx)
(b)	δ(AMA, DefaultPre, CorrectEx) $\longrightarrow$ (AA, DeleteEx)
(c)	δ(AMA, DefaultPre, SupportEx) $\longrightarrow$ (AMA, DeleteEx)
(d)	δ(AMA, DefaultPre, CorrectEx) $\longrightarrow$ (AMA, DeleteEx)
9 (a)	δ(AMA, DefaultPre, DeleteEx) $\longrightarrow$ (AA, DefaultEx) pop(Ex$_{topfamily}$), push(Ex$_{top+1}$)
(b)	δ(AMA, DefaultPre, DeleteEx) $\longrightarrow$ (AMA, DefaultEx) pop(Ex$_{topfamily}$), push(Ex$_{top+1}$)

Table 3.5.: State transitions 7, 8 and 9

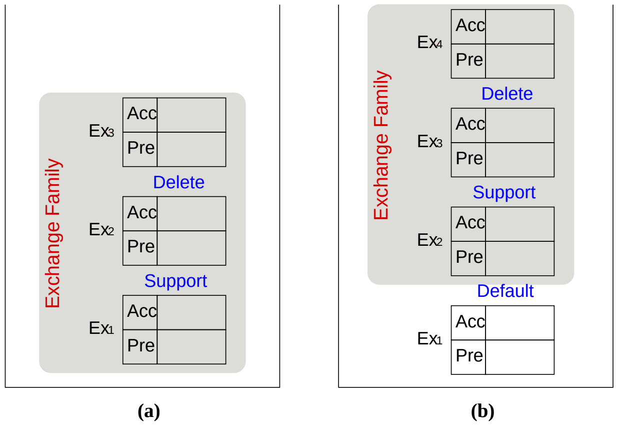


Figure 3.16.: Two cases of transition 8 (see Table 3.5) (a) The stack only contains the Exchange Family of the Ex$_{top}$. Before it is grounded, the stack is in state AA. (b) The stack contains more than the Exchange Family of the Ex$_{top}$. Before the Ex$_{top}$ is grounded, the stack is in state AMA.

10 (a)	δ(AMA, SupportPre, *anyEx*) $\longrightarrow$ (AMA, SupportEx)
(b)	δ(AMA, CorrectPre, *anyEx*) $\longrightarrow$ (AMA, CorrectEx)
11 (a)	δ(AMA, DeletePre, *anyEx*) $\longrightarrow$ (AA, DeleteEx)
(b)	δ(AMA, DeletePre, *anyEx*) $\longrightarrow$ (AMA, DeleteEx)

Table 3.6.: State transitions 10 and 11 (*anyEx* = a member of Φ)

12 (a)	δ(AMA, Acc$_{n,\theta}$, DefaultEx) $\longrightarrow$ (AMA, Ex$_{top-1}$)
(b)	δ(AMA, Acc$_{n,\theta}$, SupportEx) $\longrightarrow$ (AMA, Ex$_{top-1}$)
(c)	δ(AMA, Acc$_{n,\theta}$, CorrectEx) $\longrightarrow$ (AMA, Ex$_{top-1}$)
(d)	δ(AMA, Acc$_{n,\theta}$, DefaultEx) $\longrightarrow$ (AA, Ex$_{top-1}$)
(e)	δ(AMA, Acc$_{n,\theta}$, SupportEx) $\longrightarrow$ (AA, Ex$_{top-1}$)
(f)	δ(AMA, Acc$_{n,\theta}$, CorrectEx) $\longrightarrow$ (AA, Ex$_{top-1}$)
13 (a)	δ(AMA, Acc$_{n,\theta}$, DeleteEx) $\longrightarrow$ (AMA, Ex$_{top-1}$)
(b)	δ(AMA, Acc$_{n,\theta}$, DeleteEx) $\longrightarrow$ (AA, Ex$_{top-1}$)
(c)	δ(AMA, Acc$_{n,\theta}$, DeleteEx) $\longrightarrow$ (E, ϵ)
14	δ(AMA, Acc$_{n+1}$, DefaultEx) $\longrightarrow$ (AMA, DefaultEx)
15 (a)	δ(AMA, Acc$_{n+1}$, DeleteEx) $\longrightarrow$ (AMA, DefaultEx)
(b)	δ(AMA, Acc$_{n+1}$, DeleteEx) $\longrightarrow$ (AA, DefaultEx)

Table 3.7.: State transitions 12, 13, 14 and 15

DefaultEx is pushed (transition 9 in Table 3.5). Similar as in case of transition 8, if there were no other Exchanges on the stack than the popped Exchange Family, then the stack will return to state AA; Otherwise, its state will be AMA.

If either dialog participant creates a SupportPre or CorrectPre in state AMA, then a corresponding Exchange will be simply pushed onto the stack, independently of the current Ex$_{top}$, and the state of the stack remains AMA (transition 10 in Table 3.6). In case of an input of DeletePre, the resulting state can be either AA or AMA (transition 11 in Table 3.6), as in case of transition 8.

Of course, in the state of AMA, dialog participants can also generate Acceptance as input. If an Acc$_{n,\theta}$ is generated when a non-DeleteEx is on the top of the stack, the resulting state can be either AA or AMA (transition 12 in Table 3.7). If Ex$_{top}$ is a DeleteEx, the resulting state can also be E - in case that all the Exchanges on the stack belong to the Exchange Family of Ex$_{top}$ (transition 13 in Table 3.7). If dialog participants generate IU of type Acc$_{n+1}$, the resulting states of the transition will change, in comparison to transitions 12 and 13, because this Acceptance pushes an additional DefaultEx onto the stack (transition 14 and 15 in Table 3.7).

16	δ(*anyState*, Unqualified, *anyEx*) $\longrightarrow$ (*anyState*, *anyEx*)

Table 3.8.: State transition 16 (*anyState* = a member of Q, *anyEx* = a member of Φ)

In all the three states E, AA and AMA, an input signal of Unqualified would result in no stack operations at all (transition 16 in Table 3.8). Its only effect is that the discourse holder would, in the next turn, create a SupportPre, CorrectPre or DeletePre to either assist her dialog partner's grounding process or to cancel her effort.

Coming back to the two central questions of this section: what is the overall structure of a discourse and how IUs operate in this structure. The answer can be given now as the following:

A dialog discourse can be modeled as a push-down automaton, which is defined as a 7-tuple: W = (Q,Σ,Φ,δ,s,Ω,F) where

- Q is the set of stack state and $Q = \{$E, AA, AMA$\}$;
- Σ is the input alphabet and $\Sigma = \{$Acc$_{n,\theta}$, Acc$_{n+1}$, DefaultPre, SupportPre, CorrectPre, DeletePre, Unqualified$\}$;
- Φ is the stack alphabet and $\Phi = \{$DefaultEx, SupportEx, CorrectEx, DeleteEx$\}$;
- δ is the set of transition relations as summarized in Table 3.9;
- s is the start state and s = E;
- Ω is the initial stack symbol and Ω = DefaultPre;
- F consists of finite states and $F = \{$E$\}$.

Three general issues

When using the above push-down automaton to model a dialog, a user needs to be aware of the following three issues:

The first one is the issue of turn taking. So far, the discussion about the automaton has stayed neutral in terms of which dialog participant takes turn, i.e., the transitions are intended to be valid independently of who generates the input signal. Although it is theoretically possible for both dialog participants to generate any types of input signal at any time, some are not realistic, as also pointed out by Traum[Tra94]. For example, in case that Mary pushes a DefaultEx it is not possible for her to generate an Acc$_{n,\theta}$ or Acc$_{n+1}$ for this Exchange. From this perspective, a rule

1	δ(E, DefaultPre, ϵ) $\longrightarrow$ (AA, DefaultEx)	
2 (a)	δ(E, SupportPre, ϵ) $\longrightarrow$ (E, ϵ)	
(b)	δ(E, CorrectPre, ϵ) $\longrightarrow$ (E, ϵ)	
(c)	δ(E, DeletePre, ϵ) $\longrightarrow$ (E, ϵ)	
(d)	δ(E, Acc$_{n,\theta}$, ϵ) $\longrightarrow$ (E, ϵ)	
(e)	δ(E, AA, Acc$_{n+1}$, ϵ) $\longrightarrow$ (E, ϵ)	
3 (a)	δ(AA, DefaultPre, DefaultEx) $\longrightarrow$ (AMA, DefaultEx)	
(b)	δ(AA, SupportPre, DefaultEx) $\longrightarrow$ (AMA, SupportEx)	
(c)	δ(AA, CorrectPre, DefaultEx) $\longrightarrow$ (AMA, CorrectEx)	
(d)	δ(AA, DeletePre, DefaultEx) $\longrightarrow$ (AMA, DeleteEx)	
4 (a)	δ(AA, Acc$_{n,\theta}$, DefaultEx) $\longrightarrow$ (E, ϵ)	
(b)	δ(AA, Acc$_{n,\theta}$, SupportEx) $\longrightarrow$ (E, ϵ)	
(c)	δ(AA, Acc$_{n,\theta}$, CorrectEx) $\longrightarrow$ (E, ϵ)	
5	δ(AA, Acc$_{n+1}$, DefaultEx) $\longrightarrow$ (AA, DefaultEx)	pop(Ex$_{top}$) and push(Ex$_{top+1}$)
6 (a)	δ(AA, DefaultPre, DefaultEx) $\longrightarrow$ (AA, DefaultEx)	
(b)	δ(AA, SupportPre, DefaultEx) $\longrightarrow$ (AA, SupportEx)	
(c)	δ(AA, CorrectPre, DefaultEx) $\longrightarrow$ (AA, CorrectEx)	
7	δ(AMA, DefaultPre, DefaultEx) $\longrightarrow$ (AMA, DefaultEx)	
8 (a)	δ(AMA, DefaultPre, SupportEx) $\longrightarrow$ (AA, DeleteEx)	
(b)	δ(AMA, DefaultPre, CorrectEx) $\longrightarrow$ (AA, DeleteEx)	
(c)	δ(AMA, DefaultPre, SupportEx) $\longrightarrow$ (AMA, DeleteEx)	
(d)	δ(AMA, DefaultPre, CorrectEx) $\longrightarrow$ (AMA, DeleteEx)	
9 (a)	δ(AMA, DefaultPre, DeleteEx) $\longrightarrow$ (AA, DefaultEx)	pop(Ex$_{topfamily}$), push(Ex$_{top+1}$)
(b)	δ(AMA, DefaultPre, DeleteEx) $\longrightarrow$ (AMA, DefaultEx)	pop(Ex$_{topfamily}$), push(Ex$_{top+1}$)
10 (a)	δ(AMA, SupportPre, *anyEx*) $\longrightarrow$ (AMA, SupportEx)	
(b)	δ(AMA, CorrectPre, *anyEx*) $\longrightarrow$ (AMA, CorrectEx)	
11 (a)	δ(AMA, DeletePre, *anyEx*) $\longrightarrow$ (AA, DeleteEx)	
(b)	δ(AMA, DeletePre, *anyEx*) $\longrightarrow$ (AMA, DeleteEx)	
12 (a)	δ(AMA, Acc$_{n,\theta}$, DefaultEx) $\longrightarrow$ (AMA, Ex$_{top-1}$)	
(b)	δ(AMA, Acc$_{n,\theta}$, SupportEx) $\longrightarrow$ (AMA, Ex$_{top-1}$)	
(c)	δ(AMA, Acc$_{n,\theta}$, CorrectEx) $\longrightarrow$ (AMA, Ex$_{top-1}$)	
(d)	δ(AMA, Acc$_{n,\theta}$, DefaultEx) $\longrightarrow$ (AA, Ex$_{top-1}$)	
(e)	δ(AMA, Acc$_{n,\theta}$, SupportEx) $\longrightarrow$ (AA, Ex$_{top-1}$)	
(f)	δ(AMA, Acc$_{n,\theta}$, CorrectEx) $\longrightarrow$ (AA, Ex$_{top-1}$)	
13 (a)	δ(AMA, Acc$_{n,\theta}$, DeleteEx) $\longrightarrow$ (AMA, Ex$_{top-1}$)	
(b)	δ(AMA, Acc$_{n,\theta}$, DeleteEx) $\longrightarrow$ (AA, Ex$_{top-1}$)	
(c)	δ(AMA, Acc$_{n,\theta}$, DeleteEx) $\longrightarrow$ (E, ϵ)	
14	δ(AMA, Acc$_{n+1}$, DefaultEx) $\longrightarrow$ (AMA, DefaultEx)	
15 (a)	δ(AMA, Acc$_{n+1}$, DeleteEx) $\longrightarrow$ (AMA, DefaultEx)	
(b)	δ(AMA, Acc$_{n+1}$, DeleteEx) $\longrightarrow$ (AA, DefaultEx)	
16	δ(*anyState*, Unqualified, *anyEx*) $\longrightarrow$ (*anyState*, *anyEx*)	

Table 3.9.: State transitions in the MMPDA model

of thumb can be established: *the dialog participant who does not push the current Ex$_{top}$ onto the stack has the obligation to create the next IU.* This means, if Mary pushes an Exchange onto the stack, her dialog partner Tom would have the obligation to reply to it.

The second issue is the role of grounding criteria. Since the discourse stack is a private model of one dialog participant, the discourse holder carries out all the transitions based on her own grounding criteria. More specifically, if she categorizes the reply of her dialog partner as an input signal Acc$_{n,\theta}$ or Acc$_{n+1}$, then this means that this reply fulfills the grounding criteria of *both* dialog participants (see the discussion on grounding criteria in section 3.2.1 on page 45); If she is supposed to ground an Exchange, which is initiated by her dialog partner, she will generate an IU that only satisfies *her own* grounding criteria. Although this input signal may not be the expected Acceptance for her dialog partner, for her, it is already a valid one and should result in popping the Exchange from the stack. Of course this can cause problems if she has already popped this Exchange when her dialog partner contradicts her reply and pushes a CorrectEx onto the stack. This is the issue of other-initiated Correct Exchanges (see below).

The problem of not being able to handle other-initiated Correct Exchanges is common to models that do not model grounding as a recursive process. It can always happen that dialog participants realize that they misunderstood each other several turns ago and correct it later on. However, the corresponding grounding unit, in which the misunderstanding occurred, was already declared as grounded by one or both dialog participants and may not be available any longer. Though, a non-recursive grounding model can solve this problem by keeping several grounding units available for a period of time after they are grounded, e.g., Traum implemented the dialog discourse as a bounded stack which represents the currently accessible grounding units. In the MMPDA model, all the popped Exchanges are collected in another stack called "History" and can be reloaded to the discourse stack if needed (see the example of Conditional Relevance on page 3.3.1). Given these solutions, a non-recursive grounding process probably still requires less cognitive load than a recursive one, which needs to keep track of every utterance ever spoken in the dialog. This second stack is also important for its own reason because it records a part of the on-going discourse, namely those Exchanges that have been grounded and popped off the discourse.

Keeping these three points in mind, a dialog example is hand-coded using the MMPDA model in the next section. This example demonstrates how the push-down automaton operates when being used to model a dialog in practice.

Discourse operation from the perspective of a dialog participant

Imagine a robot called BIRON who has never talked with a human but possesses the knowledge of the MMPDA model[11]. It knows how to generate IUs and is also familiar with stack transitions. Is BIRON then able to carry out multi-modal dialog with a human? Such as that in Table 3.10?

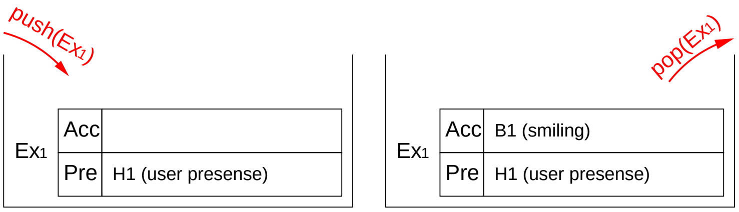
At the beginning of this dialog, BIRON first detects a human whom it views as a potential inter-

[11] Of course, it also needs other relevant language capabilities such as speech recognition, understanding and generation

H1:	*(Walks into the room where BIRON is.)*
B1:	*(Smiles friendly to the human upon detecting her.)*
H2 :	(a) Oh, hello. (b) who are you?
B2 :	(a) Hello, (b) I'm BIRON. (c) And what is your name?
H3:	(a) Tom. (b) Hm, are you a robot or something like that?
B3:	Pardon?
H4:	Are you a robot or something?
B4:	What is "something"?
H5:	Forget it.
B5:	*(looking embarrassed)*
H6:	...

Table 3.10.: A dialog example. (B: BIRON, H: human. Non-verbal behaviors are italic and included in parentheses.)

action partner. BIRON thus takes the presence of the human as an input signal of type DefaultPre (H1), which is manifested non-verbally. BIRON then pushes a corresponding DefaultEx Ex_1 onto its internal discourse stack (transition 1 in Fig. 3.17). BIRON wants to interact with the human and, therefore, addresses the Ex_1: it creates an IU and instantiates the non-verbal generator on its BLayer with a smile (B1). BIRON thinks that this IU is a valid Acceptance, which grounds the current Ex_1, and then pops it from the stack (transition 4(a)).



transition 1 **transition 4(a)**

Figure 3.17.: Dialog segment H1 - B1

Then in H2, the human is surprised by the presence of BIRON and, after intuitively greeting it (H2(a)), she asks BIRON for its identity(H2(b)). BIRON takes these two dialog acts as two input IUs that push two DefaultEx (Ex_2 and Ex_3) onto its stack in the reversed order[12] (transition 1 and 3(a) in Fig. 3.18). BIRON first addresses Ex_2 by generating an IU whose verbal generator on its BLayer is instantiated with "Hello" (B2(a)). This IU grounds the Ex_2 and BIRON pops it from the stack (transition 12(a)). Now the top Exchange on the stack is Ex_3, which BIRON needs to address. It creates an IU (B2(b)) to answer this questions and pops Ex_3 from the stack (transition 4(a)). BIRON also wants to know the user's name and, therefore, creates an IU (B2(c)) based on this motivation and pushes a new DefaultEx Ex_4 onto the stack (transition 1). Now the user

[12] Concerning the order of pushing Exchange onto the stack, see section 3.3.3

has the obligation to ground Ex_4 as she also does with IU H3(a) so that the stack is empty again (transition 4(a)).

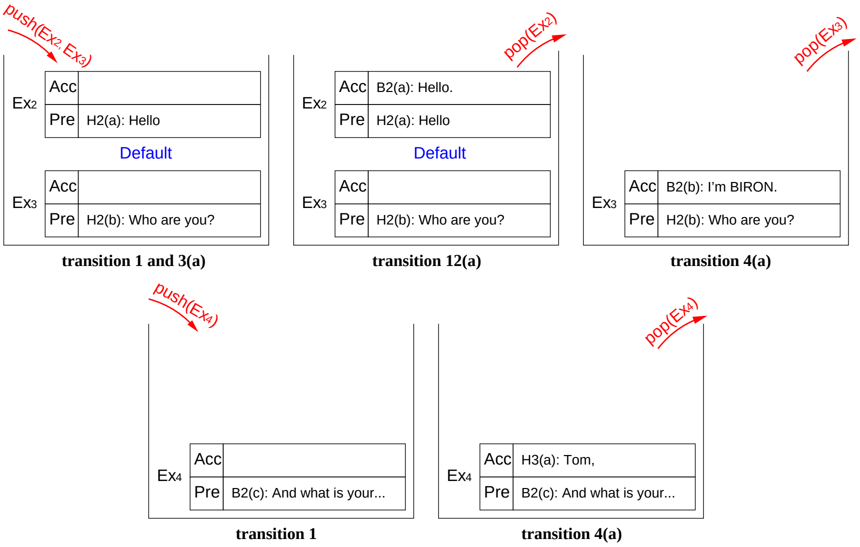


Figure 3.18.: Dialog segment H2(a) - H3(a)

The human is still not sure about BIRON's identity and initiates a question H3(b) to confirm her supposition that BIRON is a robot. BIRON pushes a corresponding DefaultEx Ex_5 onto the stack and tries to ground it (transition 1 in Fig. 3.19). However, it can not fully understand the human's IU and, therefore, creates a SupportPre (B3) and pushes Ex_6, a SupportEx, onto the stack (transition 3(b)). Then the human rephrases her IU slightly (H4), which is categorized as Unqualified by BIRON because it still can not understand the human. Following the transition 16, BIRON carries out no stack operations for the human's input[13] but creates a new SupportPre (B4), for which a new SupportEx Ex_7 is pushed onto the stack (transition 10(a)).

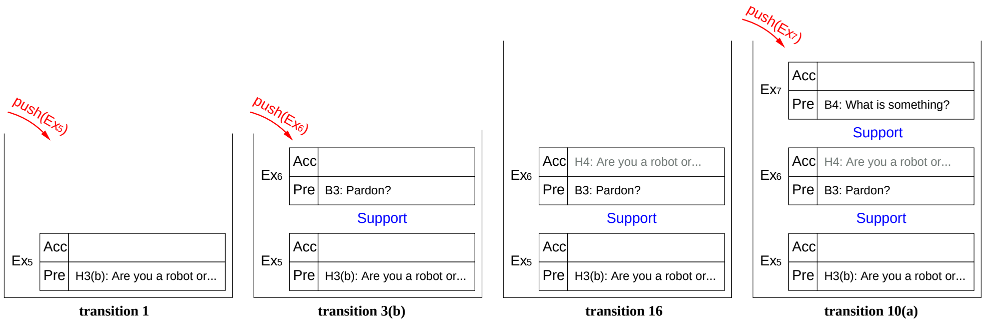
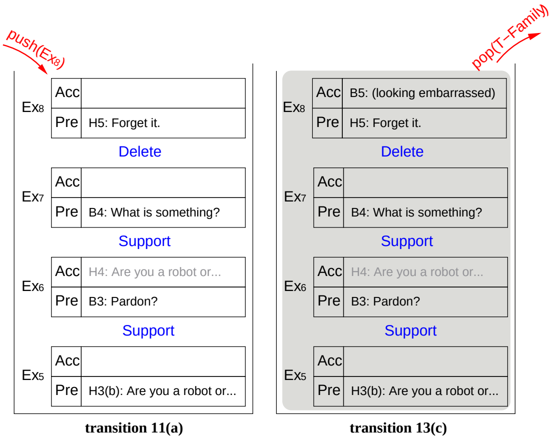


Figure 3.19.: Dialog segment H3(b) to B4

In H5, the human gives up her effort to make her question understood by BIRON and creates

[13] Originally, no stack operations at all should be carried out based on transition 16. However, for structural convenience, H4 is put into the position of Ex_6's Acceptance)

a DeletePre. Thereupon, BIRON pushes a new DeleteEx Ex_8 onto the stack based on transition 11(a) (Fig. 3.20). BIRON feels embarrassed that it can not understand the human despite repeated attempts and acknowledges the human's proposal for canceling its grounding process: it creates an IU with the motivation to demonstrate its embarrassment with an appropriate facial expression (B5) and grounds the Ex_8 with it. Based on transition 13(c), the Exchange Family of Ex_8 is also removed from the stack which is now empty.



transition 11(a) transition 13(c)

Figure 3.20.: Dialog segment H5 - B5

The operations that BIRON has carried out during this dialog are summarized in table 3.11. As can be seen, a robot should be able to carry out multi-modal dialog in the style of this example when it possesses the grounding model proposed in this chapter.

H1:		*(walks into the room where BIRON is.)*	T1: (E, DefaultPre, ϵ) $\longrightarrow$ (AA, DefaultEx$_1$);
B1:		*(smiles friendly to the human upon detecting her.)*	T4(a): (AA, Acc$_{n,\theta}$, DefaultEx$_1$) $\longrightarrow$ (E, ϵ);
H2:	(a)	Oh, hello,	T1: (E, DefaultPre, ϵ) $\longrightarrow$ (AA, DefaultEx$_3$)
	(b):	who are you?	T3(a): (AA, DefaultPre, DefaultEx$_3$) $\longrightarrow$ (AMA, DefaultEx$_2$)
B2:	(a)	Hello, .	T12(a): (AMA, Acc$_{n,\theta}$, DefaultEx$_2$) $\longrightarrow$ (AA, DefaultEx$_3$)
	(b)	I'm BIRON.	T4(a): (AA, Acc$_{n,\theta}$, DefaultEx$_3$) $\longrightarrow$ (E, ϵ);
	(c)	And what is your name?	T1: (E, DefaultPre, ϵ) $\longrightarrow$ (AA, DefaultEx$_4$)
H3:	(a)	Tom.	T4(a): (AA, Acc$_{n,\theta}$, DefaultEx$_4$) $\longrightarrow$ (E, ϵ);
	(b)	Hm, are you a robot or something like that?	T1: (E, DefaultPre, ϵ) $\longrightarrow$ (AA, DefaultEx$_5$)
B3:		Pardon? .	T3(b): (AA, SupportEx, DefaultEx$_5$) $\longrightarrow$ (AMA, SupportEx$_6$)
H4:		Are you a robot or something?	T16. (AMA, Unqualified, SupportEx$_6$) $\longrightarrow$ (AMA, SupportEx$_6$)
B4:		What is “something”?	T10(a): (AMA, SupportPre, SupportEx$_6$) $\longrightarrow$ (AMA, SupportEx$_7$)
H5:		Forget it.	T11(a): (AMA, DeletePre, SupportEx$_7$) $\longrightarrow$ (AA, DeleteEx$_8$)
B5:		*(looking embarrassed)*	T13(c): (AMA, Acc$_{n,\theta}$, DeleteEx$_8$) $\longrightarrow$ (E, ϵ)
H6:		...	

Table 3.11.: BIRON's stack operations during the dialog. (B = BIRON, H = human, T*n* = transition *n*, Non-verbal behaviors are italic and included in parentheses).

3.3. Evaluating the MMPDA model

This chapter, so far, has discussed three existing grounding models and also proposed a new grounding model, the MMPDA model, which possesses the ability of handling multi-modal dialog contributions. The practical convenience that the MMPDA model enables for the implementation will be discussed in chapter 5. For now, it is important to look at the benefits and deficiencies of this new model from a theoretical point of view. In subsection 3.3.1 and 3.3.2, dialog excerpts from a corpus recording English casual conversation and artifical dialog examples covering different types of repair behaviors are hand-modeled using the MMPDA model. The goal is to identify the range of dialog phenomena that can and cannot be handled by the MMPDA model In subsection 3.3.3, general strengths and weakness of the model are analyzed.

3.3.1. Using the London-Lund corpus

The London-Lund corpus [Tha83, Ore83, Ste84] is a collection of British casual English conversations that were clandestinely recorded in and around university settings. This corpus contains many dialog phenomena in everyday conversation and was also used by Clark [Cla92] in his proposal of the contribution model (see section 3.1.1). In this subsection the dialog excerpts used by Clark are hand-modeled using the MMPDA model.

Below, the dialog excerpts are named using Clark's definition. The discourse holder is randomly selected as dialog participant A. The original utterances in the examples are numbered for the reason of clearance. Exchanges are numbered based on *the order of their initiation*. For example, if an Exchange with grounding relation Default (DefaultEx) is pushed at the beginning of the dialog and then another Exchange with grounding relation Support (SupportEx) is pushed, the DefaultEx is denoted as *DefaultEx*$_1$ and the SupportEx as *SupportEx*$_2$, although they are Exchange of different types. The second column of each excerpt table presents the state transitions performed by each utterance. For a complete list of state transitions in the MMPDA model, see page 62.

Contribution by turns: The commonest form of contributing to a dialog is contribution by turns. The example in Table 3.12 can be easily modeled using transition 1 and 4(a) of the MMPDA model.

A1: how far is it from Huddersfield to Coventry .	T1: δ(E, DefaultPre, ϵ) $\longrightarrow$ (AA, DefaultEx$_1$);
B1: um. about um a hundred miles -	T4(a): δ(AA, Acc$_n$, DefaultEx$_1$) $\longrightarrow$ (E, ϵ);
A2: so, in fact, if you were . living in London during. that period, you would be closer - .	T1: δ(E, DefaultPre, ϵ) $\longrightarrow$ (AA, DefaultEx$_2$);

Table 3.12.: Dialog example: Contribution by turns (T = transition)

Conditional relevance: Replies of a listener may not be conditionally relevant so that the speaker has to correct her contribution. The dialog between B and A in Table 3.13 is such an example. Two utterances of this example are worth mentioning:

- Utterance A1: In the original analysis of Clark, (a) and (b) of are considered as constructing *one* utterance and A1 thus plays the role of Acceptance as a whole (compare to Fig.3.1 on page 32). From this view, A1 can be modeled using the transition 4(a) of the MMPDA model. Another possible view is that (a) and (b) are to be viewed as two distinct input signals of A, and (b) corrects or complements the content of (a). If so, the MMPDA model, as Clark's contribution model, would have difficulty to handle it, because (a) and (b) are two attempts to provide Acceptance for B1 and it is the question how to categorize (a) if (b) is the correct Acceptance for B1. However, it is more likely, that the discourse holder A considers A1 as only one single input signal. Recall transition 6 (page 58) which states that, in case of multiple IUs in one turn or self-initiated self-repair, the listener can consider all the Exchanges initiated by the speaker all at once. Parallel to this assumption, it is conceivable that a dialog participant considers Acceptance candidates, which complement or correct each other, all at once, too.
- Utterance B2: This utterance is also an interesting case because with transition 1, A reloads the DefaultEx$_1$, that was initiated by B1 and was erroneously considered as grounded by A, from the History. A does this in her discourse model because B2 is a CorrectPre which obviously addresses that Exchange (see the discussion on page 61). After performing transition 6(c) for utterance B2, the discourse holder A now has two ungrounded Exchanges on her stack, which are both initiated by B (DefaultEx$_1$ and CorrectEx$_2$). This situation is equivalent to self-initiated self-repair and transition 4(c) should be performed and the two Exchanges are grounded all at once by utterance A2.

B1: k who evaluates the property — .	T1: δ(E, DefaultPre, ϵ) $\longrightarrow$ (AA, DefaultEx$_1$);
A1: (a) uh whoever you ask((ed)),	T4(a): δ(AA, Acc$_n$, DefaultEx$_1$) $\longrightarrow$ (E, ϵ);
(b). the surveyor for the building society.	
B2: No, I meant who decides what price	*T1: δ(E, DefaultPre, ϵ) $\longrightarrow$ (AA, DefaultEx$_1$);*
it will go on the market -	T6(c): δ(AA, CorrectPre, DefaultEx$_1$) $\longrightarrow$ (AA, CorrectEx$_2$);
A2: (-snorts) . whatever people will pay ..	T4(c): δ(AA, Acc$_n$, CorrectEx$_2$) $\longrightarrow$ (E, ϵ);
	pop DefaultEx$_1$ and CorrectEx$_2$ all at once
B3: but why was Chetwynd Road so cheap —	T1: δ(E, DefaultPre, ϵ) $\longrightarrow$ (AA, DefaultEx$_3$);

Table 3.13.: Dialog example: Conditional relevance (T = transition)

Contribution within turns: Sometimes dialog participants initiate Exchanges with a "large" Presentation, e.g. telling a story. In such situations, the listener does not need to provide explicit Acceptance for each of the utterances. The example in Table 3.14 is such a case. The question here is again, whether parts of an utterance, such as (a), (b) and (c) in B1, construct one Presentation, as suggested by Clark, or multiple ones. If it is only one Presentation, then the dialog can be simply modeled using transition 1 and 4(a).

B1: (a) but you daren't set synthesis again you see, (b) you set analysis, and you can.put the answers down (c) and your assistant *examiners will work them,*	T1: δ(E, DefaultPre, ϵ) $\longrightarrow$ (AA, DefaultEx$_1$);
A1: *yes quite, yes, yes*	T4(a): δ(AA, Acc$_n$, DefaultEx$_1$) $\longrightarrow$ (E, ϵ);
B2: But if you give them a give n them a free hand on synthesis and they'd be marking all sorts of stuff, because they can't do the stuff *themselves,*	T1:δ(E, DefaultPre, ϵ) $\longrightarrow$ (AA, DefaultEx$_2$);
A2: "quite m*	T4(a): δ(AA, Acc$_n$, DefaultEx$_2$) $\longrightarrow$ (E, ϵ);
B1: I must watch [continues]	T1:δ(E, DefaultPre, ϵ) $\longrightarrow$ (AA, DefaultEx$_3$);

Table 3.14.: Dialog example: Contribution within turns (T = transition)

Installment contributions: Some input signals are not associated with sentences but with parts of them, as Clark terms. With the MMPDA model, such contributions can be easily explained: A2 pushes a SupportEx (SupportEx$_3$ in Table 3.15) that B3 grounds (and pops). Now the top Exchange on the stack is the DefaultEx$_2$ that was initiated by B2. With the information collected from the SupportEx$_3$, A3 is able to ground it.

B1: Banque Nationale de Liban —	T1: δ(E, DefaultPre, ϵ) $\longrightarrow$ (AA, DefaultEx$_1$);
A1: yes	T4: δ(AA, Acc$_{n,\theta}$, DefaultEx$_1$) $\longrightarrow$ (E, ϵ);
B2: nine to thirteen	T1: δ(E, DefaultPre, ϵ) $\longrightarrow$ (AA, DefaultEx$_2$);
A2: sorry	T3(b): δ(AA, SupportPre, DefaultEx$_2$) $\longrightarrow$ (AMA, SupportEx$_3$)
B3: nine . to . thirteen	T12(b): δ(AMA, Acc$_n$, SupportEx$_3$) $\longrightarrow$ (AA, DefaultEx$_2$)
A3: yeah [continues]	T4(a): δ(AA, Acc$_n$, DefaultEx$_2$) $\longrightarrow$ (E, ϵ);

Table 3.15.: Dialog example: Installment contribution (T = transition)

Completions: During a dialog, the Presentation of a dialog participant can be completed by her dialog partner's Presentation. For example, in Table 3.16, B1 completes A1. This means, the utterances A1 and A3 actually construct one single contribution of the discourse holder A and they are separated in the example because B barges in. While Clark ignores this fact and views A1 and A3 as two distinct input signals, the MMPDA model is able to produce a plausible explanation by imposing transition 6(a): Utterance B1 and A2 are first viewed as composing a Support Exchange (SupportEx$_2$) so that A1 and A3 are temporarily to be viewed as two input signals. However, after the SupportEx$_2$ is popped, the discourse holder A is able to perform transition 6(a) which allows both the DefaultEx$_1$ and DefaultEx$_3$ to be grounded all at once by B.

A1: um the problem is a that you(('ve)) got to get planning consent -	T1: δ(E, DefaultPre, ϵ) $\longrightarrow$ (AA, DefaultEx$_1$);
B1: before you start -	T3(b): δ(AA, SupportPre, DefaultEx$_1$) $\longrightarrow$ (AMA, SupportEx$_2$)
A2: before you start on that part, yes	T12(b): δ(AMA, Acc$_n$, SupportEx$_2$) $\longrightarrow$ (AA, DefaultEx$_1$)
A3: you can do anything internally, you wish	T6(a): δ(AA, DefaultPre, DefaultEx$_1$) $\longrightarrow$ (AA, DefaultEx$_3$)
B3: but the big stuff is, the external stuff [continues]	T5: δ(AA, Acc$_{n+1}$, DefaultEx$_3$) $\longrightarrow$ (AA, DefaultEx$_4$) *pop DefaultEx$_1$ and DefaultEx$_3$ all at once, then push DefaultEx$_4$*

Table 3.16.: Dialog example: Completions (T = transition)

As shown above, the MMPDA model is able to provide a plausible explanation for all dialog phenomena of casual dialog as selected by Clark. In comparison to his contribution model, the MMPDA model is even more sophisticated especially in case of conditional relevance and completions. As the three existing grounding models, which are discussed in section 3.1, the MMPDA model does not specify the coverage (or the size) of IUs, i.e., the basic contribution unit of dialog participants. When adopting the MMPDA model for computer applications, the definition of the IU needs to be refined based on the typical dialog patterns in the domain.

3.3.2. Modeling repair

To evaluate the ability of the model to handle conversational repair, artificial dialog excerpts are constructed based on the repair categorization criteria of Traum [Tra94].

Conversational repairs are an important mechanism to solve understanding problems during a dialog. Traum [Tra94] classifies repairs as to who causes the problem (self or other), who initiates the repair (self or other) and in which turn the understanding problem is identified. The definitions of the repair types are generally based on the pattern of dialog, in which the speaker holds the first turn and the listener replies in the second turn, then the speaker utters something again in the third turn and the listener replies it in the fourth turn. Table 3.17 summarizes the commonest repair types.

In the following, artificial repair dialogs are hand-modeled using the MMPDA model. Since the examples of the last section already revealed the challenges of long utterances, the dialog examples in the following are only constructed with short utterances that are typical in the home tour scenario. Note, these examples are *not* intended to cover all the possibilities of repair dialog, instead, they should convey the idea as to how the repair phenomena are viewed and modeled using the MMPDA model.

Turn	Repair	Definition
first turn	self-initiated self-repair	The speaker repairs her own utterance without a prompting from another participant.
second turn	other-initiated self-repair	After the listener addresses the speaker's problem in the second turn, the speaker repairs her own utterance in the third turn.
	other-initiated other-repair	The listener notices the problem of the speaker's utterance and repairs it in the second turn.
third turn	third-turn repair	Based on the listener's second-turn reply, the speaker realizes that she is misunderstood and repairs the listener in the third turn.
fourth turn	fourth-turn repair	Based on the speaker's third-turn reply, the listener realizes that she misunderstood the speaker and repairs it herself in the fourth turn.

Table 3.17.: Conversational repair

Self-initiated self-repair: In the example in Table 3.18, dialog participant A first asks B to go to the kitchen with (a) and then corrects herself with (b). As transition 6(c) specifies, B addresses both Exchanges in correlation with each other and only provides one final Acceptance for both Exchanges.

A1: (a) Go to the kitchen,	T1: δ(E, DefaultPre, ϵ) $\longrightarrow$ (AA, DefaultEx$_1$);
(b) I mean the living room.	T6(c). δ(AA, CorrectPre, DefaultEx$_1$) $\longrightarrow$ (AA, CorrectEx$_2$)
B1: OK.	T4(c): δ(AA, Acc$_n$, CorrectEx$_2$) $\longrightarrow$ (E, ϵ);
pop DefaultEx$_1$ and CorrectEx$_2$ all at once	

Table 3.18.: Self-initiated self-repair

Other-initiated self-repair: Based on the MMPDA model, other-initiated self-repair is the case in which the listener pushes an Exchange with the grounding relation Support (the SupportEx$_2$ in Table 3.19) to acquire more information about the speaker's Exchange (DefaultEx$_1$). Such cases can be modeled using transition 3(b) and 12(b).

A1: Go to the kitchen.	T1: δ(E, DefaultPre, ϵ) $\longrightarrow$ (AA, DefaultEx$_1$);
B1: Kitchen?	T3(b): δ(AA, SupportPre, DefaultEx$_1$) $\longrightarrow$ (AMA, SupportEx$_2$)
A2: It is the room in which we cook.	T12(b): δ(AMA, Acc$_n$, SupportEx$_2$) $\longrightarrow$ (AA, DefaultEx$_1$)
B2: OK.	T4(a): δ(AA, Acc$_n$, DefaultEx$_1$) $\longrightarrow$ (E, ϵ);

Table 3.19.: Other-initiated self-repair

Other-initiated other-repair: In other-initiated other-repair in Table 3.20, the listener pushes a CorrectEx (CorrectEx$_2$) and corrects the speaker's utterance directly.

A1: Go to the kitchen.	T1: δ(E, DefaultPre, ϵ) $\longrightarrow$ (AA, DefaultEx$_1$);
B1: You mean the living room.	T3(c): δ(AA, CorrectPre, DefaultEx$_1$) $\longrightarrow$ (AMA, CorrectEx$_2$)
A2: Oh yes.	T12(c): δ(AMA, Acc$_n$, CorrectEx$_2$) $\longrightarrow$ (AA, DefaultEx$_1$)
B2: OK.	T4(a): δ(AA, Acc$_n$, DefaultEx$_1$) $\longrightarrow$ (E, ϵ);

Table 3.20.: Other-initiated other-repair

Third-turn repair: In the example of third-turn repair in Table 3.21, B misunderstood A1. The discourse holder A recognizes B1 as an input of type Unqualified and carries out the transition 16. Then, in the third turn, A pushes a CorrectEx (CorrectEx$_2$) onto the stack, which is supposed to be grounded by B2. In the current example, B grounds the CorrectEx$_2$ explicity (Acc$_n$ in transition 12(c)) and then go to the kitchen. B2(b) is a sufficient evidence that A is correctly understood and A, therefore, takes it as B's Acc$_n$ for her initial DefaultEx$_1$ and pops this Exchange from her stack.

A1: Go to the kitchen.	T1: δ(E, DefaultPre, ϵ) $\longrightarrow$ (AA, DefaultEx$_1$);
B1: OK. (going to the living room)	T16: δ(AA, Unqualified, DefaultEx$_1$) $\longrightarrow$ (AA, DefaultEx$_1$)
A2: The kitchen, I said!	T6(c). δ(AA, CorrectPre, DefaultEx$_1$) $\longrightarrow$ (AA, CorrectEx$_2$)
B2: (a) Oh, sorry. (going to the kitchen) *pop DefaultEx$_1$ and CorrectEx$_2$ all at once*	T4(c): δ(AA, Acc$_n$, CorrectEx$_2$) $\longrightarrow$ (E, ϵ)

Table 3.21.: Third-turn repair

Fourth-turn repair: In the example in Table 3.22, B first misunderstood A1, but both dialog participants are not aware of the problem. As the discourse holder A proposes A2, B realizes the problem and pushes SupportEx$_3$ onto the stack (in B2). After clarifying this with A (B2 and A3), B is able to execute the task correctly (B3). Note, it is also possible that A is never made aware of the problem, e.g., if B does not initiate the clarification question (B2) but simply says "OK" and corrects itself by heading to the kitchen. In this situation, the discourse holder A would simply think that B accepts her two instructions (A1 and A2) without any problems.

A1: Go to the kitchen.	T1: δ(E, DefaultPre, ϵ) $\longrightarrow$ (AA, DefaultEx$_1$);
B1: OK. (going to the living room)	T4(a). δ(AA, Acc$_n$, DefaultEx$_1$) $\longrightarrow$ (E, ϵ);
A2: And turn on the oven.	T1: δ(E, DefaultPre, ϵ) $\longrightarrow$ (AA, DefaultEx$_2$);
B2: Oh, You want me to go to the kitchen?	T3(b): δ(AA, SupportPre, DefaultEx$_2$) $\longrightarrow$ (AMA, SupportEx$_3$)
A3: Yes.	T12(b): δ(AMA, Acc$_n$, SupportEx$_3$) $\longrightarrow$ (AA, DefaultEx$_2$)
B3 OK. (going to the kitchen)	T4(a). δ(AA, Acc$_n$, DefaultEx$_2$) $\longrightarrow$ (E, ϵ);

Table 3.22.: Fourth-turn repair

As can be seen from the examples above, many of the conversational repair can be modeled with Exchanges of Support or Correct grounding relations. Combining the following three factors: *repairer* (who initiates the Support or Correct Exchange), *repairee* (who initiated the Exchange that needs to be repaired) and *timing* (whether the repair Exchange is created before or after the listener replies in the second turn, the most important repair types can be systematically modeled using the concept of Exchange, as summarized in Table 3.23.

Repairer	**Repairee**	**Timing**	**Repair Type**
speaker	speaker	before	self-initiated self-repair
speaker	speaker	after	other-initiated self-repair
speaker	listener	before	*(not possible)*
speaker	listener	after	third-turn repair
listener	speaker	before	*(not possible)*
listener	speaker	after	other-initiated other-repair
listener	listener	before	*(not possible)*
listener	listener	after	fourth turn repair

Table 3.23.: Modeling repair with Exchanges. (before/after = the repair Exchange is initiated before or after the listener replies in the second turn)

3.3.3. General benefits and Deficiencies

In the MMPDA model, the grounding process is non-recursive. This structure avoids the problem that the grounding process can not be ended properly, which is the case in the contribution model of Clark [Cla92] and the exchange model of Cahn and Brennan [CB99, Cah92]. This advantage is similar to that of the finite state model of Traum [Tra94]. The major difference between the MMPDA model and Traum's is that in the MMPDA model, repair is *not* viewed as a part of the grounding unit[14]. Instead, it constructs a new grounding unit with certain grounding relation to the current grounding unit. This solution is more flexible because it allows "partial grounding" of an account and can explain why repeated grounding is necessary in some situations (compare to the deficit of Traum's model in section 3.1.2 on page 36).

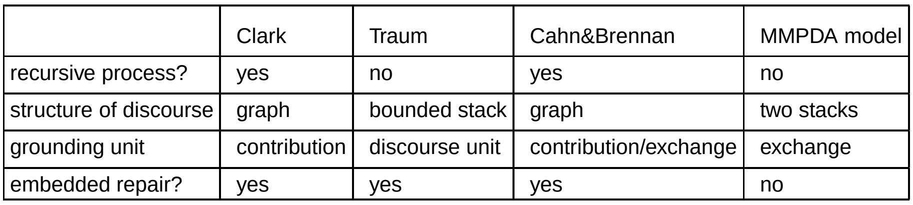
As to the issue of multi-modality, the structure of Interaction Unit is an attempt to bridge the gap between two strands of multi-modality modeling. Unlike Cassell's [Cas00] architecture, the MMPDA model does not require categorization of different types of information and thus simplifies the dialog mechanism and increases the implementability of the model. . To account for more

[14] Recall that the grounding unit in Traum's model is the discourse unit and in the MMPDA model, it is the Exchange

sophisticated embodied interaction, the MMPDA model can be easily extended in two ways. The first one is extending the Behavior Layer with a “Modality Manager”, which accomplishes sophisticated modality fusion and selection. This local extension will not affect the overall dialog management. The second way to extend this model is to add synchronization mechanism into the Behavior Layer to enable a synchronized behavior generation. Last but not least, the IUs can also be used to extend dialog systems with a simple discourse management mechanism (e.g., finite state-based) since it only affects the representation of local interaction contributions. Given that the majority of the current dialog systems running on robot systems are finite state-based (section 2.4), the concept of IUs would enable many other robots to handle multi-modal input and output without changing the underlying interaction management mechanism.

The structure of a stack in its original sense is quite inflexible because of its last-in first-out principle (LIFO). This inflexibility has the consequence for the order in which Exchanges should be processed. In most cases, it is sufficient to assume that dialog participants first address the top Exchange on the stack, then those below it. However, in some other cases, it is not easy to determine this order. For example, in case of multiple DefaultPre in one turn. Should these IUs be pushed in the order as they are created or in the reversed order? Pushing them in their original order means that the listener should address the last DefaultPre first, which is often untrue. But it is not always the case either that the listener addresses the first DefaultPre first which means that the DefaultPre are pushed in the reversed order. In fact, which IU the listener first addresses is not only regulated by the “mechanical” means of a stack, but also by the salience of individual IUs. For example, if the listener has difficulty to understand one of the IUs, it is likely that she first addresses this one before others. A possible extension of the MMPDA model is, therefore, to relax the order of pushing and popping Exchanges, which may result in new scientific questions as to how to handle the grounding relations if Exchanges are not connected in a fixed order and the Mother-Son-relationship may not be clear.

Since the concept of IU is based on the reciprocal nature of behaviors involved in an embodied interaction, its strength lies in the modeling of information that is intended to address the other dialog participant. Some applications, however, focus on the development of subtle human-like behaviors from which no clear motivations relating to other participants can be derived, e.g., looking away when thinking about something. Although the MMPDA model can still be used (e.g., by instantiating non-verbal generator on the Behavior Layer with “looking away”), it can not explain why these behaviors should be generated and, therefore, may not be the best choice



	Clark	Traum	Cahn&Brennan	MMPDA model
recursive process?	yes	no	yes	no
structure of discourse	graph	bounded stack	graph	two stacks
grounding unit	contribution	discourse unit	contribution/exchange	exchange
embedded repair?	yes	yes	yes	no

Table 3.24.: Differences between the dialog models

for such applications.

3.4. Summary

In this chapter a novel computational model of multi-modal grounding, the MMPDA model, was proposed that was motivated by the existing works of Clark, Traum and Cahn&Brenann. From the perspective of grounding, the new model combines the advantages of Traum's non-recursive structure of grounding and Cahn&Brennan's concept of Exchanges and, thus, avoids their main problems. As to the capability of handling multi-modality, this model makes use of evocative functions of both verbal and non-verbal behaviors involved in an interaction and extends the definition of common ground. By representing dialog contributions with Interaction Units, the MMPDA model is able to naturally handle multi-modality using the grounding mechanism. In the last section of evaluation, the MMPDA model was evaluated with dialog examples from the literature and it turned out that the model is able to provide plausible explanation for many dialog phenomena. Further, the MMPDA model was compared to the existing works. As can be seen, the new model is a powerful interaction management mechanism because it both improves the grounding mechanism for dialog management itself and extends it so that the concept of grounding can cover more aspects of an embodied interaction.

So far, the discussion has been on a theoretical level and the implementability of this model can not yet be proven. The next chapter will address this issue and present the implemented Interaction Management System for the Bielefeld Robot Companion.

+

4. Developing the Interaction Management System for BIRON

The MMPDA model proposed in the previous chapter was implemented for the Interaction Management System of the robot BIRON, the Bielefeld Robot Companion (Fig.4.1), which is a research platform for HRI studies. As the main interaction interface of BIRON, the Interaction Management System plays a crucial role in facilitating task execution, enabling social behaviors and increasing usability of the entire robot system. This chapter provides a detailed account of the implementation platform and scenario as well as the technical realization of the Interaction Management System.

This chapter is organized as follows: In section 4.1, the hardware platform and the software infrastructure of BIRON are described briefly. Then, in section 4.2 the implementation scenario "home tour" is discussed. Details about the technical realization of the Interaction Management System are presented in section 4.3

4.1. The implementation platform: BIRON

In the following, the hardware base of BIRON and its software architecture including the most important modules of the system are briefly described.

4.1.1. Hardware

BIRON is based on a *Pioneer PeopleBot*^{TM} of MobileRobots Inc. (formerly ActivMedia Robotics, LLC) and is equipped with a number of sensors (Fig. 4.1). In the following the most important sensors are listed:

- *Pan-Tilt Camera*: The Sony EVI-D31 camera mounted at a hight of 142 cm of the robot is equipped with a pan-tilt unit that allows a motor-driven steering of horizontally 100 degrees and vertically 25 degrees. This camera is used to aquire images of user's face and upper body.
- *Stereo microphones*: Two AKG C 400 BL microphones are mounted right below the touch screen and they are responsible for receiving speech signals from users.

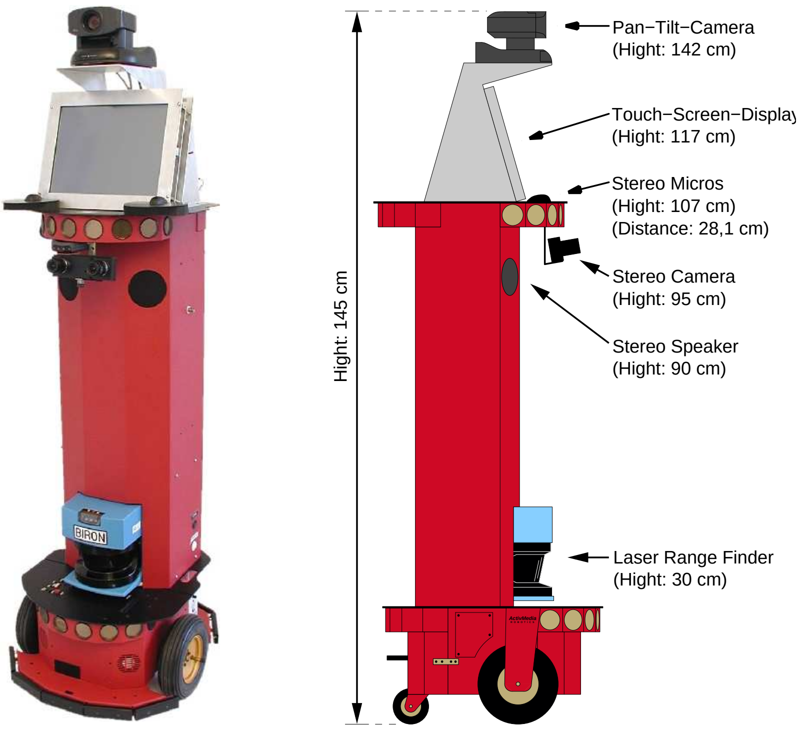


Figure 4.1.: BIRON: the Bielefeld Robot Companion

- *Stereo Camera*: The Videre Design STH-MDCS stereo camera is mounted at a hight of 95 cm and acquires images of users' gestures.
- *Laser Range Finder*: The SICK laser range finder measures the distance to objects in the close surroundings (180 degrees range of maximally 32 m). It is used to detect human legs for BIRON.

The signals received by these sensors are forwarded to various software modules that analyze them to extract symbolic meanings from the signals. For example, if the laser range finder detects two human legs and the pan-tilt camera a human face, then it is possible that a human is standing in front of the robot. Subsequently, this symbolic information of human presence is transfered to other software modules that make decisions as to what to do with this information given the current interaction situation. In short, in order to make use of the signals, the robot system needs two types of software modules: reactive and deliberative modules that perform signal-level and symbol-level analysis. In the section below, the most important ones among them are presented.

4.1.2. Software

BIRON is a highly complex system consisting of more than 35 software modules that need to be organized in a meaningful way. This section first depicts the architecture of BIRON that provides a “frame” in which the modules are arranged according to their reactive or deliberative nature. Then, the most important modules are presented. Finally, the general communication and cooperation principles between these modules are discussed.

Software modules performing different tasks in an integrated system have to be put into a meaningful technical context so that their operation can be coordinated in a flexible manner. For this purpose, a powerful three-layered, hybrid architecture [KFS04, FKH+05, Kle05] (Fig. 4.2) was developed for BIRON. Here, the reactive and the deliberative modules are located on the Reactive and Deliberative Layer, respectively. The Execution Supervisor on the Intermediate Layer coordinates the data transfer between these modules and is the heart of the entire system. The central issue of the architecture is *to ensure a flexible shift of control over system behaviors*. More specifically, as a situated computer device a robot can not be controlled only by deliberative modules that make high-level decisions based on user’s input, it should also be able to react to unexpected environmental changes. For example, if the robot detects obstacles on its way to the kitchen, where it is expected by the user, it should be able to “know” that the need to avoid these obstacles is more urgent than following the shortest path to the kitchen, and act accordingly. Here, the control over the robot’s behavior is shifted from the deliberative modules, which made the decision to go to the kitchen, to the reactive modules, which have the direct control of the hardware to adapt its speed. To ensure timely and appropriate control shift, the Execution Supervisor operates based on a finite state-machine that represents different operation contexts as different states. In certain states, commands from the deliberative or reactive modules are to be rejected because of urgent needs of other modules. The Execution Supervisor thus possesses the central control of *most* software modules that perform “cross-layer” operations, i.e., modules that have to coordinate with modules on other layers[1]. In the following, the most important such modules (the darkly shaded ones in Fig. 4.2) are briefly described.

Person Attention System (PAS): In order to start an interaction with human users, a robot must be able to recognize a human. Then, during the interaction, it should be able to recognize whether the communication partner is attending to it, which is particularly important when several persons are around and they may be talking to each other. These abilities are preconditions for a successful interaction and are the responsibilities of the module PAS [LKH+03, FKL+04, Lan05]. The approach adopted here is multi-modal person tracking and attention control: The system moves the pan-tilt camera around to detect human faces, uses two microphones for sound source localization, and the laser range finder for leg detection. Based on the analysis of these percepts and the combination of them, the PAS makes the decision as to whether a human exists in BIRON’s vicinity and whether she intends to interact with BIRON. Several assumptions facilitate the decision making process. For example, to identify human’s interaction intention, the system considers the following combination of percepts: If legs are not moving (the person is not walking), a face can be detected and it is gazing in the direction of the robot (she is facing the robot), and sound can be detected from the same direction as the legs and the face (the person is speaking),

[1] There are exceptions: a small number of modules communicate with each other directly although they are located on different layers. This is to reduce the time loss caused by data transfer between modules. For example, the module Person Attention System has direct communication channels to the Speech Recognizer and to the Interaction Management System.

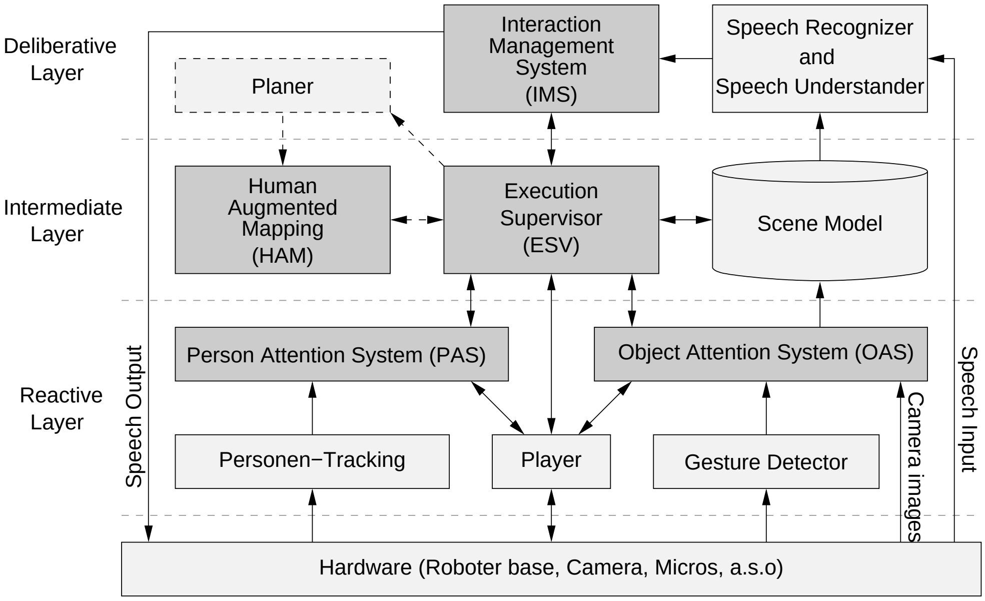

Deliberative Layer
Intermediate Layer
Reactive Layer
Planer
Interaction Management System (IMS)
Speech Recognizer and Speech Understander
Human Augmented Mapping (HAM)
Execution Supervisor (ESV)
Scene Model
Person Attention System (PAS)
Object Attention System (OAS)
Personen−Tracking
Player
Gesture Detector
Speech Output
Camera images
Speech Input
Hardware (Roboter base, Camera, Micros, a.s.o)

Figure 4.2.: Architecture of BIRON, adapted from [KFS04]

then she probably is talking to BIRON. Once a human is identified as intending to interact with BIRON, the PAS concentrates its multi-modal perception acquisition on this person. Only then, other modules of BIRON become active and the entire system is able to interact with the person and carry out tasks. Further, during the interaction, the PAS can steer the motor of BIRON to follow the user on demand. Here, the legs of the human are the most important percept.

Object Attention System (OAS): This System [HHFS05, LHW+05, Haa07] is responsible for acquiring images of objects that are pointed to by users. When the interaction situation requires it, as determined by the Interaction Management System, this system takes over the control of the pan-tilt camera from the PAS and steers it to the direction of the gesture, which is detected by a Gesture Detector. From the pictures that are acquired by the camera the system extracts the image of the object that is pointed to. Subsequently, this image is stored in a memory, the so-called Scene Model. Also other information on the object such as type and owner are stored here, if they are provided by the user (in her utterance).

Human Augmented Mapping (HAM): For a mobile robot it is crucial to have a spatial model that enables it to orient itself when moving around. Ideally, the spatial model should correspond to that of humans to facilitate communication. The module HAM [TC06, THCSE06, SLW+07] facilitates this communication by establishing connections between the semantic dimension and the topological dimension of the robot's spatial model. More specifically, when receiving messages from the Interaction Management System that the current location is called, e.g., "kitchen", the HAM assigns this label to the current topological mark in its map. With this knowledge, BIRON is able to "know" where is the kitchen if it needs to navigate there to perform some tasks.

Speech Recognizer and Understander: The Speech Recognizer [Fin95] can recognize distance speech that is recorded by the two microphones mounted on BIRON as well as be used in combination with close-talking microphones that human users wear. The major challenge for the Speech Understander [HW06, HWS06, Hue07b] is to deal with spontaneous speech that is often ungrammatical. The approach adopted here is based on bottom-up frame merging technique: Each word is defined as an instance of a frame, a kind of top-level category, and the semantic meaning of an utterance is acquired by merging frames of individual words.

Interaction Management System(IMS): The MMPDA model discussed in the previous chapter was implemented in the IMS of BIRON. It is the central interaction module of BIRON and will be discussed in detail in sections 4.3 and the chapter 5.

The above software modules are connected to each other within the so-called XML Communication Framework [FKH+05, WFBS04], short: XCF. In this framework, each module communicates with other modules by sending and receiving messages in the Extensible Markup Language (XML) [Xml]. More specifically, when the PAS detects a human who is intending to interact with BIRON, it activates the Speech Recognizer to process speech. The resulting parts of speech are forwarded to the Speech Understander which constructs a semantic representation of the speech and sends them to the IMS. Based on the proposition of the speech, the IMS either replies to the user directly or sends commands to the ESV. The ESV changes its state and/or forwards the command to other modules, e.g., to the OAS, PAS or HAM. When these modules finish their processing they send their results back to the ESV that forwards them to the IMS. Based on these results, the IMS generates output as reply to the user. This output is constructed using text blocks that are pre-specified in a configuration file. With the open-source speech synthesizer Festival [Fes] the output is converted from text to speech. This entire process is illustrated in Fig. 4.3.

Besides reacting to user-initiated speech, BIRON can also take interactional initiatives, e.g., the IMS takes conversational initiatives during the interaction based on the needs of grounding. Such initiatives are triggered by the IMS itself and do not involve other modules of BIRON. Initiatives involving other modules usually originate from the PAS which observes the environment constantly during the interaction (in contrast, the OAS is only activated when it is needed). Once a noticeable event occurs, e.g., the current interaction partner leaves or there are obstacles around, the PAS informs the IMS of these events. the IMS then considers its own states and generates appropriate speech output. When the ESV is also informed, then the state of the entire BIRON system is changed, too. This process is illustrated in Fig. 4.4.

Note, Fig. 4.3 and 4.4 only roughly illustrate the general cooperation principles between modules in BIRON. In the course of the Implementation-Evaluation-Cycles of the IMS (see chapter 5), some aspects of these principles were modified. More specifically, in the second Implementation-Evaluation-Cycle, the control over the speech input is transfered from the PAS to the IMS (page 103) and the IMS also generates non-verbal output with a life-like character displayed on the touch screen of BIRON (page 105).

The software modules implemented on the platform BIRON enable the robot to detect and fol-

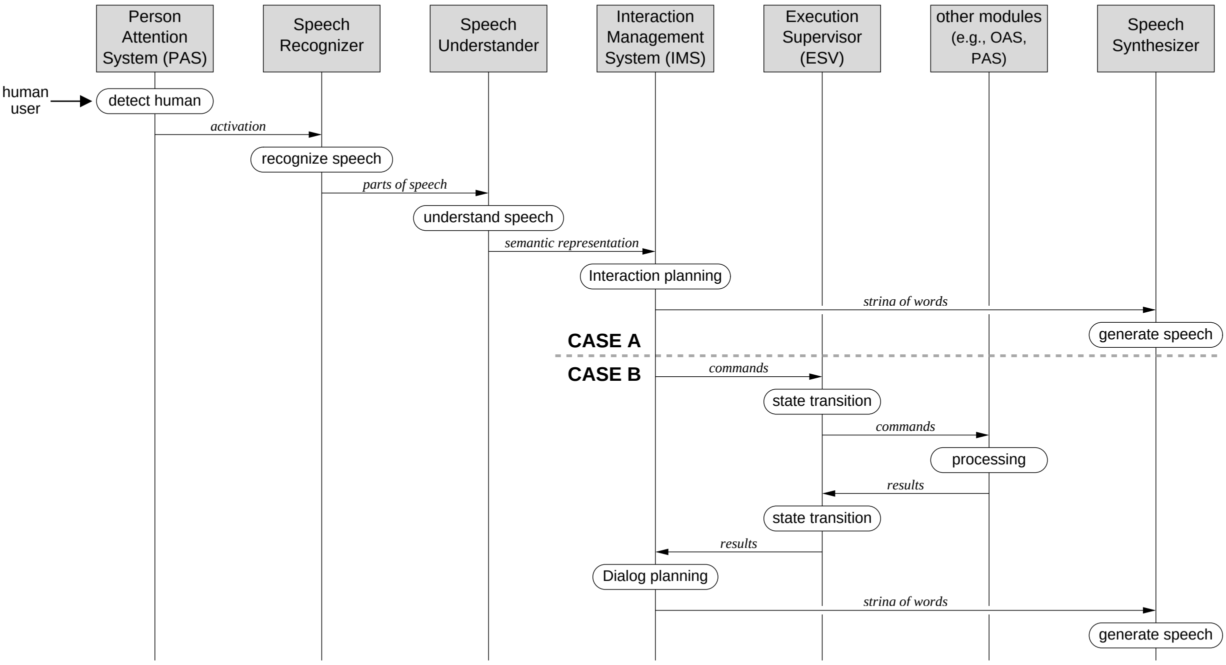


Figure 4.3.: The cooperation between modules in BIRON. Case A: The IMS directly replies to user; Case B: The IMS accesses other modules before replying (OAS = Object Attention System).

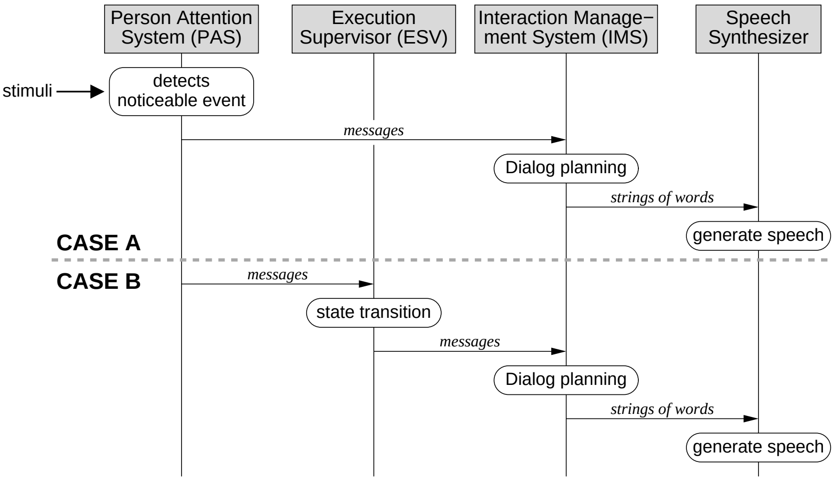


Figure 4.4.: Taking initiative based on environmental changes. Case A: The PAS directly informs the IMS of the environmental changes; Case B: The PAS informs the IMS via ESV.

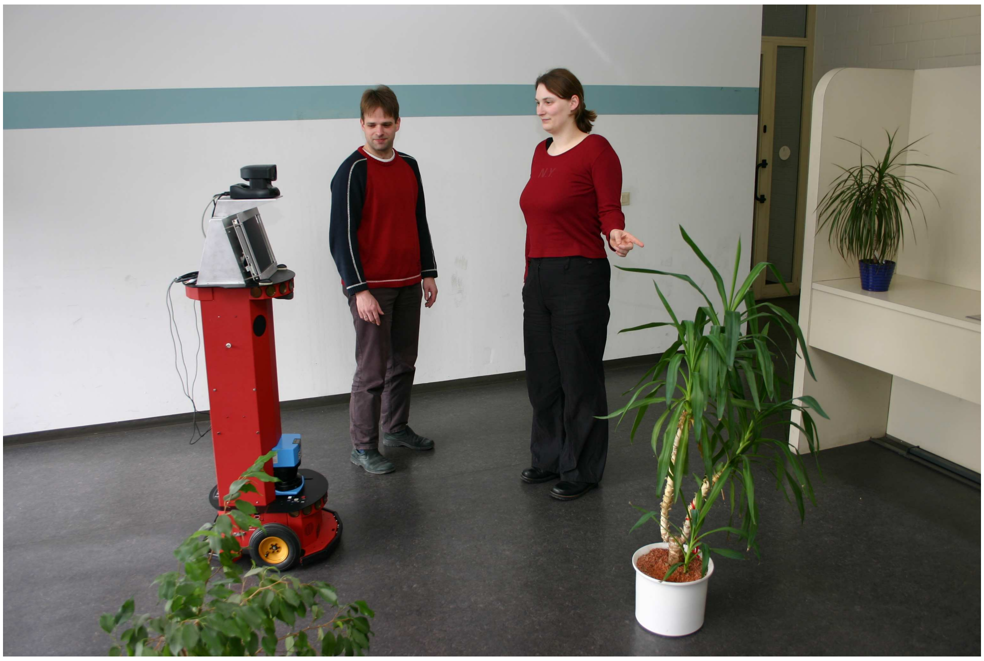


Figure 4.5.: The home tour scenario

low persons, carry out dialog with human users, track their deictic gestures, focus on objects pointed by users, store collected multi-modal information into a memory, and remember names of locations. These abilities are needed in the implementation scenario that is discussed below.

4.2. The implementation scenario: home tour

Within the European Union project COGNIRON (The Cognitive Robot Companion [Cog04]), a key experiment named *home tour* is specified as implementation scenario for BIRON. The basic idea is that, after a user bought a new robot from a shop, she shows it her home to prepare it for future tasks. To realize this scenario, the robot should be mobile, interactive and possess a high standard of perceptual capabilities. More specifically, BIRON should be able to follow the user around, and when she points to an object, e.g., a cup, and says “This is my favorite cup.”, the robot should be able to understand the user’s speech, track her deictic gesture, detect the object that the user is pointing to and remember its features, i.e., name, color, images, et cetera. Similarly, if the user says “This is the kitchen”, BIRON should associate the name “kitchen” with a topological mark in its map and “remember” the location this way. With this knowledge, BIRON is able to, e.g., navigate to the kitchen, fetch the cup and use it to perform some tasks. Figure 4.5 illustrates an example setup of the scenario.

The challenge of this scenario for most reactive modules of BIRON, e.g., PAS, OAS, Gesture Detector, Speech Recognizer and so on lies in the complexity and the ambiguity of the real home environment. Firstly, they need to decide whether to start the processing at all. Taking the Speech Recognizer as an example: In the real home environments sound can come from various sources. Beside general background noises, human speech can also come from TV set, radio or conversations between humans who are not involved in the interaction with BIRON. Under this circumstance, to determine what sound should be processed as the speech of the current user is a highly challenging task. Secondly, the reactive modules need to correctly recognize relevant features from the environments, which is difficult given the unstructuredness of everyday environments. For example, when the PAS attempts to detect human legs using the laser ranger finder, the assumption is that a pair of obstacles of reasonable width standing in a reasonable distance to each other indicate the existence of human legs. But such information can be highly ambiguous when there are desks or chairs around, the legs of which have similar width. Last but not least, the behavior of users also has great influence on processing results of reactive modules, e.g., how they stand, in which direction they gaze, how they point to an object, in what speed they do it, and so on. Since it is not realistic to ask users to wear special sensors in their everyday life, which could often improve the reliability of processing results, many unconscious user behaviors can cause failure of reactive modules, too.

The home tour scenario also poses new scientific questions to the deliberative module IMS. When collecting multi-modal information that was previously unknown to the robot, the success of task execution relies on correct processing of reactive modules to a great extent. This means that the IMS has little a-priori knowledge to do sophisticated top-down reasoning. The consequence is that the IMS can *not* directly affect the task-related performance of the robot, as in many desktop applications. In this situation, what functions or behaviors should be implemented in the IMS to improve the overall interaction quality becomes the main challenging issue for the implementation of the IMS. Therefore, the development paradigm for interactive systems *Implementation-Evaluation-Cycle* was adopted. The idea is to implement functions and behaviors iteratively based on insights gained in evaluations. After the basic infrastructure of the IMS was established, various functions and behaviors were implemented within two such cycles. Before presenting these behaviors in chapter 5, however, it is necessary to first look at the technical realizations of the basic infrastructure of the IMS in general, which clarifies many important technical issues of the system.

4.3. Technical realization of the Interaction Management System

The IMS is BIRON's main interaction interface to users and the MMPDA model proposed in the previous chapter was implemented in it. This section addresses the following four questions: What is the architecture of the IMS (section 4.3.1), how does the IMS interpret user speech input and determine their effects on the grounding status of the system (section 4.3.2), how is the most important configuration file MeaningEffectMatch.xml used (section 4.3.3) and what is the general processing flow of the IMS (section 4.3.4).

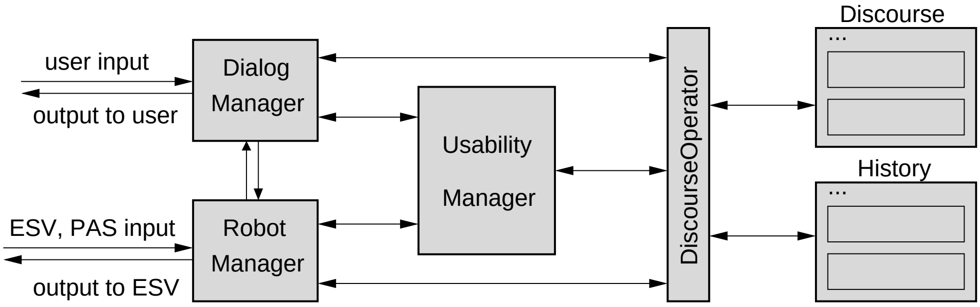


Figure 4.6.: Architecture of BIRON's Interaction Management System (ESV = Execution Supervisor, PAS = Person Attention System

4.3.1. The architecture

As illustrated in Fig. 4.6, the IMS is composed of four main components: *DialogManager*, *RobotManager*, *UsabilityManager* and *DiscourseOperator*. The discourse of the on-going interaction is represented using a stack, the so-called *Discourse*. And the interaction history is stored in another stack called *History*.

The Dialog- and RobotManager receive input from the user of different modalities. The DialogManager mainly receives the semantic representations of user's speech input, which are delivered by the Speech Understander. The RobotManager receives other information on the user from other software modules of the robot. For example, the PAS provides information as to whether the user is facing the robot and the OAS (indirectly) informs the RobotManager of what direction the user is pointing to. The communication between these modules and the RobotManager is usually coordinated by the ESV. This means that the ESV often serves as "postman" and forwards processing results of other modules to the RobotManager depending on the context. However, there is also a direct communication channel between the PAS and the RobotManager. Through this channel, the PAS periodically sends information to the RobotManager so that it is kept up-to-date with the attention of the current user.

After receiving input, the Dialog- and RobotManager process it base on the current system state. For this purpose, they contact the DiscourseOperator. The DiscourseOperator has direct access to the Discourse as well as the History and possesses the latest information on the grounding state of the system. Based on this information and the received input, the Dialog- and RobotManager make decisions as to what to do in the next step, i.e., how to manipulate the Discourse and the History. The manipulation is directly performed by the DiscourseOperator and is based on the state transitions specified in the MMPDA model. The upshot of this manipulation is always the generation of an Interaction Unit (IU), whether a Presentation or an Acceptance should be created to initiate or ground an Exchange. This IU represents the IMS' reaction to the user input: the Behavior Layer generates output to the user or the Motivation Layer sends commands to other software modules of the robot. Both the Dialog- and the RobotManager are able to initiate the creation of such an IU.

Note, the DialogManager is the main component that performs sophisticated interpretation of user input (see next subsection for more details). The RobotManager primarily serves as the interface of the IMS to other software modules of BIRON, e.g., the ESV or the PAS. During an interaction, the RobotManager interprets messages from those modules in terms of whether the execution of tasks are successful, which means whether they match DialogManager's expectations. However, these messages sometimes also cause the RobotManager to directly call the DiscourseOperator. This happens when the PAS detects noticeable events in the environment and the IMS should adjust its behaviors, as discussed in section 4.1.2.

Both Dialog- and RobotManager can call the DiscourseOperator via the UsabilityManager. This component takes care of the realization of cooperative interactive behaviors. The Usability-Manager was added in the second Implementation-Evaluation-Cycle. Details about the realized behaviors by this component can be found in section 5.2.

As can be seen, the architecture of the IMS is modular and separates domain reasoning from the grounding process: The domain-dependent decisions are made in the Dialog- and RobotManager while the DiscourseOperator is only in charge of manipulating the Discourse and the History based on the MMPDA model. If this system is to be ported to another robot system, only the Dialog- and RobotManager need to be modified. To further increase the flexibility of the system, many decisions are out-sourced to configuration files, instead of being hard-coded in the source code of the program. Section 4.3.3 shows such an example.

4.3.2. Interpreting grounding effects of speech input

Additionally to the architecture presented above, rules have to be established as to how to interpret user speech input. This responsibility includes the following two aspects: (1) How to interpret users' speech in terms of their meanings in the domain, and (2) how to determine the grounding effects of these domain contributions of the user. The Speech Understander and IMS are responsible for the first and the second aspect, respectively. In the following, the relevant rules that are applied by these two modules are discussed.

The Speech Understander attempts to classify speech input into 10 *categories* based on their semantic meanings (the first column in Table 4.1). These categories are defined following the principle of dialog acts. The semantic representation of each speech input that is sent to the IMS is marked with one of these categories. Upon receiving it, the DialogManager of the IMS

Categories	**Groups**
instruction, query, description, manipulation	Independent
correction, deletion	Dependent
confirmation, negation, object, fragment	Related

Table 4.1.: Classification of speech input

determines its effects on the grounding state of the system in two steps:

Firstly, the DialogManager examines its membership of one of the three *groups* (the second column in Table 4.1) and represents it as an IU with one of the different “roles”: Members of the Independent Group propose new tasks and the DialogManager represents them as IUs that play the role of a Presentation and initiate a new DefaultEx[2]. For members of the Dependent Group, the DialogManager initiates Delete- or CorrectEx with the corresponding IUs as Presentation. Members of the Related Group can only be responses to system’s Presentation and are, therefore, classified as potential Acceptance candidate of the current top Exchange in the Discourse.

Secondly, the DialogManager determines the effect of the new IU on existing Exchanges of the Discourse. For IUs that are created based on members of the Independent and Dependent Groups, the DialogManager calls the DiscourseOperator to carry out appropriate state transitions based on the MMPDA model. IUs for Related Group members are further examined taking into account the current interaction context. The goal is to decide (1) whether they are really an expected Acceptance, and (2) what consequence the circumstance has. Since this decision has to be made from case to case and is sometimes also based on heuristics, it is out-sourced to a XML configuration file, *MeaningEffectMatch.xml*. In the next subsection, the usage of this file is demonstrated based on two interaction excerpts.

4.3.3. The usage of the configuration file MeaningEffectMatch.xml

Consider the interaction excerpts in Table 4.2 and 4.3, which show two behaviors of the IMS. User’s utterance U2 is identified as Acceptance for BIRON’s utterance B1 in both excerpts. However, in excerpt 4.2, the user accepts B1 with a confirmation, while she does this with a negation in excerpt 4.3. Although in both cases BIRON carries out the same state transition, T12(e), it has different consequences for the subsequent interaction because of the different propositions of the two Acceptance.

In excerpt 4.2, the Acceptance U2 confirms BIRON’s assumption that the user wants it to do something that it can not do at that moment. Under this circumstance, BIRON decides to ground the initial Default Exchange (DefaultEx$_1$) with utterance B2 by carrying out transition T4[3]. Immediately after that, BIRON pushes a new Default Exchange (DefaultEx$_3$) to inform the user what she should do achieve her goal given the current system state. These two transitions enable BIRON to generate sufficient feedback to the user, which is an important mechanism to handle user operation errors. In excerpt 4.3 however, the negation of the user in utterance U2 lets BIRON realize that it misunderstood the user. Based on this understanding, BIRON carries out transition T3(c), which pushes a Correct Exchange (CorrectEx$_3$). After the user accepts this

[2] Support Exchanges that are initiated by users are not considered in the implementation.

[3] Usually, BIRON grounds an user-initiated Exchange only if it successfully executes the specified task. However, even if the execution of the task is not possible, the message that the DialogManager generates to inform the user of the problem still addresses the user’s Presentation and is, therefore, considered as Acceptance, too. This kind of Acceptance is called “Pseudo-Acceptance”.

Exchange with utterance U3, BIRON retries to ground the initial Default Exchange DefaultEx$_1$. This Exchange is updated with the information contained in U3 and BIRON succeeds in grounding it properly.

U1: Follow me.	T1: δ(E, DefaultPre, ϵ) $\longrightarrow$ (AA, DefaultEx$_1$)
B1: Do you want me to follow you?	T3(b): δ(AA, SupportEx, DefaultEx$_1$) $\longrightarrow$ (AMA, SupportEx$_2$)
U2: **Yes.**	**T12(e): δ(AMA, Acc$_{n,\theta}$, SupportEx$_2$) $\longrightarrow$ (AA, DefaultEx$_1$)**
B2: Sorry, I can't do that right now.	**T4: δ(AA, Acc$_{n,\theta}$, DefaultEx$_1$) $\longrightarrow$ (E, ϵ)**
B3: You need to first...	T1: δ(E, DefaultPre, ϵ) $\longrightarrow$ (AA, DefaultEx$_3$)

Table 4.2.: User accepts BIRON's Presentation with a confirmation. (U = user, B = BIRON)

U1: Follow me.	T1: δ(E, DefaultPre, ϵ) $\longrightarrow$ (AA, DefaultEx$_1$)
B1: Do you want me to follow you?	T3(b): δ(AA, SupportEx, DefaultEx$_1$) $\longrightarrow$ (AMA, SupportEx$_2$)
U2: **No.**	**T12(e): δ(AMA, Acc$_{n,\theta}$, SupportEx$_2$) $\longrightarrow$ (AA, DefaultEx$_1$)**
B2 How can I help you?	**T3(c): δ(AA, CorrectPre, DefaultEx$_1$) $\longrightarrow$ (AMA, CorrectEx$_3$)**
U3 I want to show you something!	T12(f): δ(AMA, Acc$_{n,\theta}$, CorrectEx$_3$) $\longrightarrow$ (AA, DefaultEx$_1$)
B3 OK, I'm looking.	T4: δ(AA, Acc$_{n,\theta}$, DefaultEx) $\longrightarrow$ (E, ϵ)

Table 4.3.: User accepts BIRON's Presentation with a negation. (U = user, B = BIRON)

As can be seen, even if the same transition is to be performed in a given interaction context, the different propositional information contained in interaction contributions affects the selection of the next transitions. The determination of these effects varies from case to case and needs to be pre-specified in some form for the implementation. To increase the flexibility of the IMS, this specification is coded in a configuration file, the so-called MeaningEffectMatch.xml. Figure 4.7 shows the segment of this file that specifies the corresponding Acceptance and next transitions for the above interaction excerpts.

In the file MeaningEffectMatch.xml, interaction context is represented as a combination of the purpose of the current top Exchange on the Discourse (attribute "currentPurpose" in tag "match") and the purpose of its Mother Exchange (attribute "motherPurpose"). In tag "acc", speech input of certain group, category or (propositional) content is defined as a legal Acceptance for the given interaction context. The tag "nextTransition" specifies the next transition that should be performed subsequently. For example, the first specification means: Given the interaction context that the current Exchange initiates a clarification question and the mother Exchange states that the user's command currently can not be executed, then the user input of category "confirmation" is the Acceptance of the current Exchange, and transition 4(a) should be performed after the current Exchange is grounded. When starting the IMS, the system parses the MeaningEffectMatch.xml into an internal structure. During the interaction, the DialogManager makes decisions based on the specifications in this structure. The advantage of doing so is that these domain-related specifications do not need to be hard-coded in the source code of the program so that they can be easily modified without changing much in the source code.

```
<match currentPurpose = "adt_confirmation_question" motherPurpose = "Cmd_cur_impossible">
    <acc>
        <group>none</group>
        <category>confirmation</category>
        <content>none</content>
    </acc>
    <nextTransition>
        <transitionNumber>4</transitionNumber>
        <transitionSubnumber>a</transitionSubnumber>
    </nextTransition>
</match>

<match currentPurpose = "adt_confirmation_question" motherPurpose = "Cmd_cur_impossible">
    <acc>
        <group>none</group>
        <category>negation</category>
        <content>none</content>
    </acc>
    <nextTransition>
        <transitionNumber>3</transitionNumber>
        <transitionSubnumber>c</transitionSubnumber>
    </nextTransition>
</match>
```

Figure 4.7.: Specification of Acceptance and next transition in a given interaction context: an excerpt from the MeaningEffectMatch.xml

Note, the file MeaningEffectMatch.xml only specifies cases in which system-initiated Exchanges expect Acceptance of type P$_n$ (see introduction of Acceptance types on page 46), i.e., the DialogManager expects the user to address the current Exchange directly. However, not all Exchanges initiated by the IMS need to be grounded in this explicit way, especially when the Dialog- or RobotManager make general comments (see section 5.1.1) or generate Exchanges only to give users some feedback for usability reasons (see section 5.2.1). In such situations, the IMS expects Acceptance of type P$_{n+1}$ or θ from the user. Such Exchanges are implemented as follows: both managers let the DiscourseOperator pop these Exchanges from the Discourse and push them onto the History immediately after they are created. In the implementation, such Exchanges are called *Ew/oA* (Exchange without Acceptance).

4.3.4. General processing flow in the DialogManager

In order to account for different interactional and technical needs, the IMS is implemented in a flexible way so that it behaves differently when being started with different parameters. However, the behavior variation mainly regulates BIRON's verbosity for different purposes (see next section) and the general processing flow of the system (see the UML activity diagram in Fig. 4.8) stays the same, as discussed below.

Upon receiving multi-modal input from a user, the DialogManager represents it as the content of the Behavior Layer (BLayer) of an Interaction Unit (IU) and analyzes it. If the user's motivation on the Motivation Layer (MLayer) of the IU can not be recognized through this analysis, the DialogManager considers this IU as initiating an ungroundable Exchange and calls the DiscourseOperator to push this user-initiated Exchange onto the Discourse. Immediately after that, the DialogManager creates an IU for itself and initiates a non-Default Exchange (e.g., a Support or Correct Exchange) with it to resolve the issue. This system-initiated Exchange is also pushed onto the Discourse.

If the motivation on the MLayer of the user IU can be recognized, the DialogManager makes the decision as to whether it initiates a new Exchange or should be viewed as a potential Acceptance. For this decision the DialogManager considers the group membership of the input (see section 4.3.2) and the current top Exchange on the Discourse.

If the user IU initiates an Exchange, the DialogManager tries to ground it by creating an IU and sending commands to other modules of BIRON on the MLayer of this IU. Upon receiving satisfying reply from other modules, the BLayer of this IU generates output as reply to the user and the user-initiated Exchange is considered as grounded. Then the DialogManager calls the DiscourseOperator to pop this Exchange from the Discourse and push it onto the History. If the Exchange initiated by the user can not be grounded, then the DialogManager initiates non-Default Exchanges to clarify the issue.

If the user IU is a potential Acceptance, then the DialogManager consults the configuration file MeaningEffectMatch.xml, as discussed above, to determine whether it is really an Acceptance. If so, the DialogManager considers the currently system-initiated top Exchange on the Discourse as grounded and removes it from the Discourse. If the user IU is not the expected Acceptance, the DialogManager initiates a new non-default Exchange to resolve the issue and pushes it onto the Discourse.

Sofar, the technical realization of the IMS has been discussed. Two advantages of the system are evident: (1) the clear separation of the grounding process from domain-related decisions in the architecture, and (2) the flexible specification of grounding effects of input as a configuration file. This technical convenience greatly simplified the implementation of various interactive behaviors in the course of the two Implementation-Evaluation-Cycles, as discussed in the next chapter.

4.4. Summary

In this chapter, technical details concerning the development of the Interaction Management System of the robot BIRON for the home tour scenario were discussed. The system was developed in a way that domain- and system-specific information is out-sourced to separate program parts or configuration files to increase the flexibility and extendability. This system architecture is beneficial for the development methodology of implementation-evaluation-cycles, as will be shown in the next chapter.

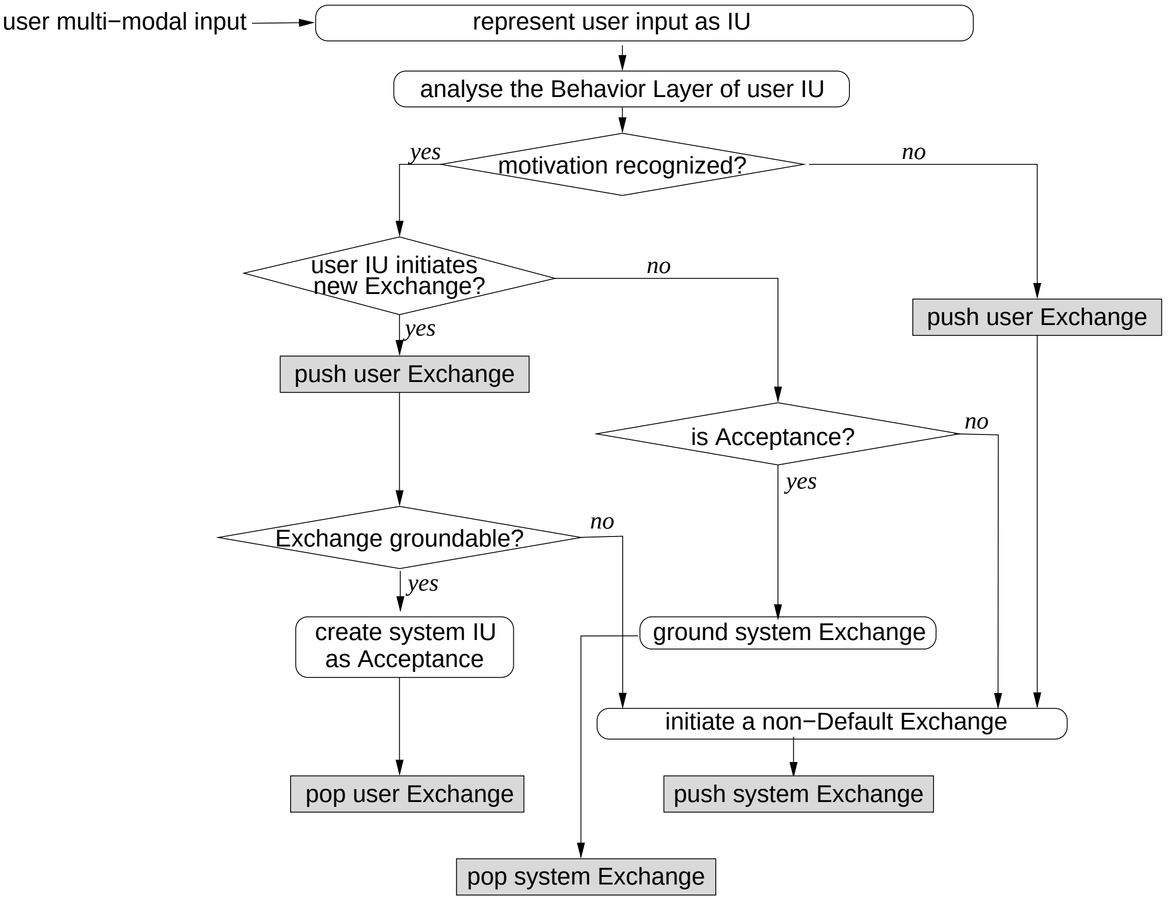


Figure 4.8.: General behavior control in the IMS (IU = Interaction Unit)

5. Implementing interactive behaviors in cycles

As mentioned in chapter 1, the development of complex multi-modal interactive behaviors for robot companions is still a young field. It is often not clear at the beginning of the development what behaviors a robot companion should exhibit to account for acceptability. Facing this challenge, the concept of Implementation-Evaluation-Cycle (IEC) was adopted. This concept is a part of the so-called *iterative development cycle* [Hul99, ADB04, Hue07a] for HCI application development. The basic idea is that neither the implementation nor the evaluation of an interactive system should be the end of the development process. Instead, they should be carried out in a cycle (Fig. 5.1): Functions are implemented and evaluated (often in form of user studies) in the first cycle, then the results from the evaluation are drawn upon for the implementation in the second cycle, and so on. This concept ensures that users are sufficiently involved in the development process and their needs can be systematically taken into consideration in the implementation of an interactive system.

The development of multi-modal interactive behaviors in the IMS went through two IECs. The focus of the first IEC (section 5.1) were functions and behaviors that facilitate domain task execution and exhibit social awareness. These behaviors were evaluated in a first user study. The observations from this evaluation served as the motivation for the focus of the second IEC (section 5.2): increasing usability. Various new functions and behaviors were developed in this IEC that was completed with a second user study. The results of this user study outline the possible focus of future work for the IMS, which would start a third IEC. In the following, these two studies are presented in the structure: goal definition, method, results, observation and discussion.

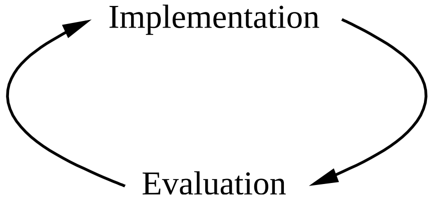


Figure 5.1.: Implementation and Evaluation Cycle (IEC)

5.1. The first IEC: facilitating domain task execution and exhibiting social awareness

The foci of the first IEC were determined to directly account for characteristics of HRI, as discussed in chapter 1: Firstly, the IMS should fulfill the most basic function of an interaction management system, i.e., facilitating domain task execution. Secondly, the IMS should enable interactive social behaviors to increase acceptability of BIRON. After the presentation of these functions and behaviors, the user study that was conducted to evaluate them is discussed.

5.1.1. Implementation

This section shows how the IMS facilitates the execution of one of the most important domain tasks in the home tour, the resolution of multi-modal object references, and how the system enables social behaviors.

Facilitating domain task executions

The most challenging domain task that the IMS has to accomplish in the home tour is to correctly handle multi-modal input of users, especially in case of deictic gestures accompanying speech such as a description “This is a cup.” The solution of the IMS (see Fig. 5.2) is based on the concept of Interaction Units (IUs) in the MMPDA model (see section 3.2.2). Recall that an IU is a two-layered structure consisting of a Motivation Layer (MLayer) and a Behavior Layer (BLayer). A verbal and a non-verbal generator are located on the BLayer, which are responsible for generating spoken language and non-verbal behaviors according to the motivation conceived on the MLayer, respectively. The relationship between the verbal and non-verbal generator can be, according to Iverson et al. [ICLC99], reinforcement, disambiguation and adding-information.

In the IMS, the DialogManager represents user input with an IU whose verbal generator on the BLayer is instantiated with an utterance, e.g., “This is a cup.” Since the input is of category “description”, this IU is considered to be a Presentation that initiates an Exchange. To provide Acceptance for this Presentation, the DialogManager first analyzes its BLayer to find out the content of its MLayer. The result of this analysis is that, in the verbal generator, what the pronoun “this” refers to is not clear. Given that the Speech Understander indicates a possible involvement of a gesture, the DialogManager decides to further analyze the non-verbal generator on the user’s BLayer, which may provide more information to disambiguate the content of the verbal generator. For this purpose, the IMS activates the OAS (Object Attention System) by sending a command to the ESV (Execution Supervisor), which performs a corresponding transition and forwards the command to the OAS. The OAS consults with the Gesture Detector and orients the pan-tilt camera on BIRON towards the position of the user’s hand. Then, in the current camera view, the OAS starts to search for a cup.

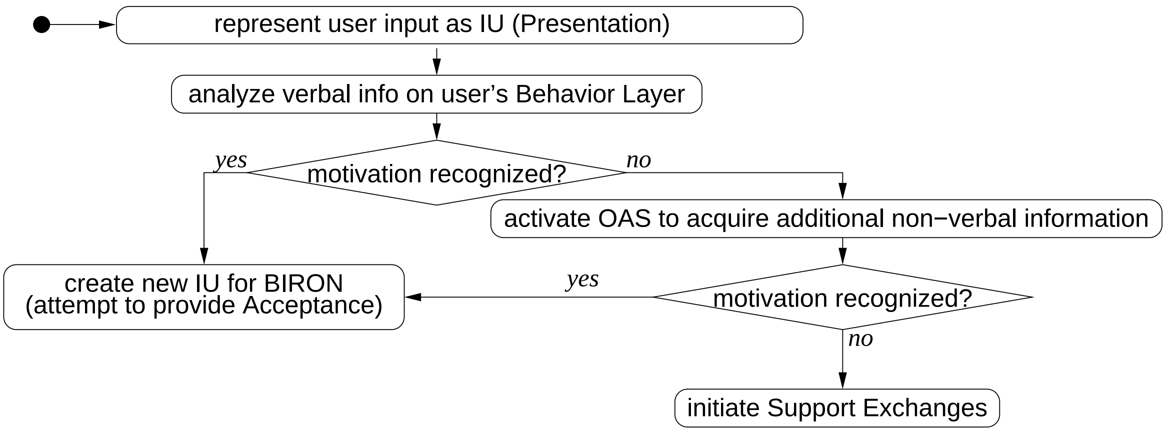

represent user input as IU (Presentation)
analyze verbal info on user's Behavior Layer
yes
motivation recognized?
no
activate OAS to acquire additional non−verbal information
create new IU for BIRON (attempt to provide Acceptance)
yes
motivation recognized?
no
initiate Support Exchanges

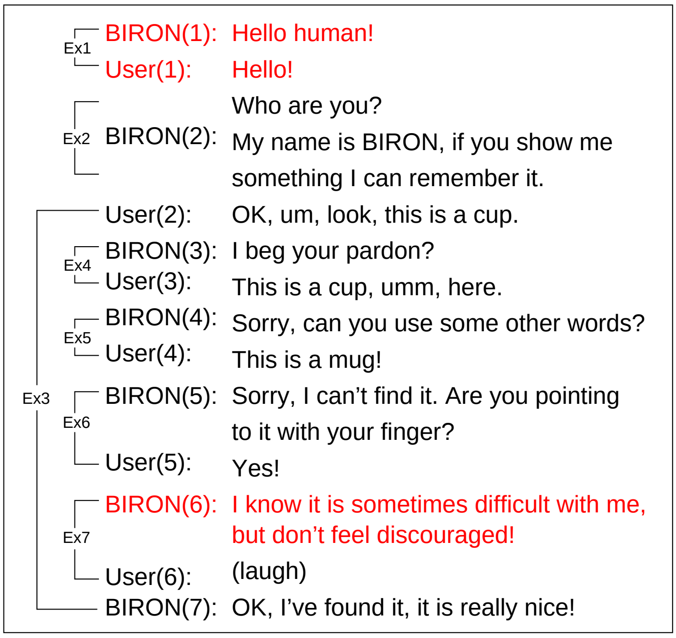
Figure 5.2.: Resolving multi-modal object references (OAS = Object Attention System)

If the OAS succeeds in finding an object next to the user's hand, it stores the image of this object and its symbolic name "cup" into BIRON's Scene Model. Afterwards, the IMS is informed of the result and can complete the analysis of the user's BLayer by concluding that the user intended to draw BIRON's attention to a cup. Since BIRON has located the cup, the task is viewed as executed. As the Acceptance of BIRON, the IMS generates an IU. In this IU, the verbal generator is instantiated with an utterance such as "OK, I've seen it." This IU then grounds the user initiated Exchange and the DialogManager of the IMS activates the DiscourseOperator to remove it from the Discourse. If the OAS has difficulty to find the cup because, e.g., no gesture can be detected, then the IMS asks a clarification question such as "Are you pointing to it with your finger?" Such clarification questions are realized as new Exchanges with Support relation to the current Exchange, which was initiated by the user. For an example of such a interaction, see Fig. 5.3.

Enabling social behaviors

The home tour scenario is not only a learning scenario for BIRON, but also a learning scenario for the user: she probably interacts with a complex robot system for the first time. Similarly to human-human interaction, such a first contact is often "tricky" but also important. The impressions of BIRON that human users gain during the home tour can affect their general opinion on BIRON. Within this context, two initiative taking behaviors were implemented in the IMS that enable BIRON to demonstrate social awareness, as presented below.

Taking greeting initiative: After a user bought a robot and unpacked it at home, she may have difficulty to start the interaction with BIRON because of her lack of knowledge of how to operate it. In this situation, it may help the user to overcome this difficulty if BIRON can greet the user on its own. This behavior would imply that the robot now works, has already perceived the user, and can also speak human language. Probably the user would reply to this greeting



BIRON(1): Hello human!
User(1): Hello!
Who are you?
BIRON(2): My name is BIRON, if you show me something I can remember it.
User(2): OK, um, look, this is a cup.
BIRON(3): I beg your pardon?
User(3): This is a cup, umm, here.
BIRON(4): Sorry, can you use some other words?
User(4): This is a mug!
BIRON(5): Sorry, I can't find it. Are you pointing to it with your finger?
User(5): Yes!
BIRON(6): I know it is sometimes difficult with me, but don't feel discouraged!
User(6): (laugh)
BIRON(7): OK, I've found it, it is really nice!

Figure 5.3.: An excerpt from an interaction between a human user and BIRON (Ex = Exchange)

intuitively as they do in human-human interaction (This assumption is actually confirmed in the user study reported in the next section). The implementation of this initiative is quite simple: Once the PAS (Person Attention System) detects a human in its vicinity it sends a message to the IMS, which then initiates a Default Exchange to greet her with utterance BIRON(1) in Fig. 5.3. Now the user is expected to provide acceptance that she heard and understood BIRON which is usually done by her reply to BIRON's greeting (User(1)). Here, the user also asks an additional question about the identity of BIRON. This question is classified as user's initiative to create a Default Exchange (Ex2) that BIRON should ground by answering the question (BIRON(2) in Fig. 5.3). If the user does not answer BIRON's greeting the IMS would remove this self-initiated Exchange from the Discourse after a pre-defined time and, thus, cancel the expectation that the user would reply.

Making remarks on its own performance: Many of BIRON's modules carry out computationally expensive processing or are subject to environmental conditions. In a real user-BIRON interaction, this means that there is a variety of factors that can negatively influence the general performance of BIRON but users know nothing about it. It may help to reduce user frustration if BIRON has the ability to show that it is also aware of its own problems and feels sorry about it. Based on this assumption the performance evaluation behavior was implemented for BIRON. The IMS realizes this behavior by counting the number of Support Exchanges it has initiated for the current topic. The Support Exchanges are only created if BIRON can not provide Acceptance for user's Presentation or her reply does not fulfill IMS's expectation. The amount of Support Exchanges, therefore, has direct correlation to the overall interaction quality. Default Exchanges have similar functions: the more Default Exchanges are created during an interaction, the better is the interaction quality because the user and BIRON can proceed to another topic only if the current one is grounded (or deleted). Based on this performance indication BIRON makes remarks to motivate users. As to the frequency of these remarks, heuristic rules were drawn on: BIRON makes remarks if it has to create at least three Support Exchanges for one topic or ground three Default Exchanges in succession. In the interaction example in Fig. 5.3, BIRON can not understand the user's utterance twice (Ex4, Ex5) and then can not find the object specified by the user (Ex6), which results in the creation of three Support Exchanges by the IMS in total. For this poor performance the IMS initiates an Ew/oA (Exchange without Acceptance) with utterance BIRON(6) to motivate the user. If it was a positive evaluation result BIRON would say something like "You are really good at working with me." The wording style of these remarks is selected randomly from a set of 3 pre-defined sentences.

After the above functions and behaviors were implemented, a user study was conducted to evaluate them, as presented below.

5.1.2. Evaluation

To test the impact of the social behaviors, BIRON was configured to either demonstrate both behaviors (extrovert BIRON) or neither of them (basic BIRON). These two types of BIRON were contrasted in a between-subject design in a user study.

Goal definition

The impact of the implemented behaviors had to be defined in a way that it could be operationalized. For the user study, the following three questions were identified as the most important:

1. Are the different verbal behaviors of the two types of BIRON perceived as different?
2. Does the different perception of BIRON's verbal behaviors have effects on the perception of other features of BIRON such as its overall performance and interaction style?
3. Does the different perception of BIRON's verbal behaviors have effects on the subjects' emotional status?

Method

Fourteen subjects aged from 20 to 37 were recruited from the Bielefeld University. The task that they were supposed to perform was showing BIRON objects lying on a desk (Fig. 5.4). In this experiment, BIRON was essentially immobile. Subjects were divided into two groups: 7

of them interacted with the basic BIRON (Group B) and the other 7 with the extrovert BIRON (Group E). Each member of the two groups was asked to interact with BIRON twice. In the first run they only received minimal instruction: they should ask BIRON what it can do and then make BIRON to do it. This means, in this run, the subjects interacted with BIRON without any knowledge about BIRON's technical limitations and language capabilities. Neither were subjects aware of the purpose of the interaction. In the second run, the subjects were asked to interact with BIRON again after they viewed a demonstration video in which a developer performed an "ideal" interaction with BIRON. Besides, they also received an example dialog that exemplified an "ideal" dialog between BIRON and a user. Altogether, each subject interacted with BIRON more than 7 minutes and the first run usually took one or two minutes more than the second run. After the experiment the subjects were asked to fill out a questionnaire.

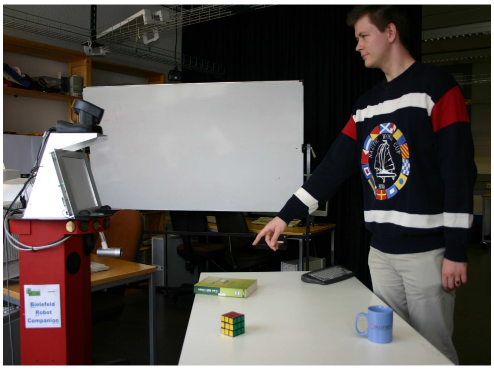


Figure 5.4.: The setup of the user study in the first IEC

To answer the first goal question subjects were asked to rate BIRON's personality. As also assumed by [WDK+05], different behaviors of a robot can cause subjects to perceive the robot as having different personalities. Four selected personality traits were used for the purpose. They are derived from the personality dimension *introversion vs. extroversion* that was proposed by Eysenck [EE75]. This dimension can be more easily associated with visible behaviors and, thus, is more suitable for the study than his two other dimensions (*neuroticism vs. emotional stability* and *psychoticism*). The 4 traits are thoughtful vs. talkative, peaceful vs. responsive, quiet vs. active, and reserved vs. impulsive. The subjects rated BIRON's personality for each of the 4 traits using a 5-point Likert scale, e.g, 1 is very thoughtful and 5 is very talk-active. For the user study, it is sufficient to assume that the higher value a selected item in the Likert scale is, the higher is the tendency of the subject to classify BIRON's personality as extrovert.

To answer the second goal question four performance problems of BIRON were listed that occur most frequently: BIRON loses the subject during the interaction, BIRON does not understand subject's utterances, the dialog with BIRON is restricted to a relatively small vocabulary and BIRON does not look in the direction of the subject's gesture. Subjects were asked to rate for each of these problems their degree of annoyance in a 5 point Likert scale. The associated question in the questionnaire was "How annoying were the following technical problems for you?" Additionally, we also asked users if they think the interaction style realized on BIRON is intuitive.

To measure users' emotional status after the interaction with BIRON they were directly asked whether they like BIRON or not.

Results

In the chart in Fig. 5.5 the result of the BIRON personality question is illustrated, the x-axis represents the personality tendency of BIRON rated by the subjects and the y-axis indicates the number of subjects who selected the corresponding items for the four traits. In Group B, subjects interacted with the basic BIRON and most of them thought BIRON tends to be introvert. In the Group E, the social behaviors of BIRON did lead to a clearly different picture than in Group B: extroversion is the most perceived personality tendency of BIRON even if the result is more distributed than in case of Group B. For a better understanding of the results, Table 5.1 summarizes the average values for the perceived personality of BIRON in both groups, which are derived from the their rating values for the four traits.

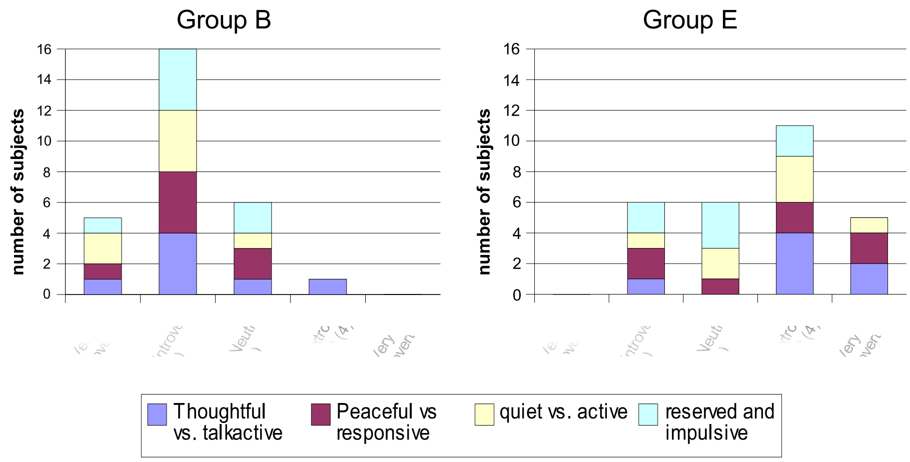


Figure 5.5.: The result of the question: "What do you think about the personality of BIRON?"

	very introvert	introvert	neutral	extrovert	very extrovert
Group B	1.25	4	1.5	0.25	0
Group E	0	1.5	1.5	2.75	1.25

Table 5.1.: Average rating results of subjects concerning BIRON's personality

Figure 5.6 shows the results of the question concerning subjects' annoyance level when they are confronted with BIRON's technical problems. Here, a slight difference in subjects' general annoyance degree can be recognized. It can be even said that members of Group B seem to be generally more angry about the technical problems than those in Group E, which is demonstrated even clearer in their average values for technical problems in Table 5.2.

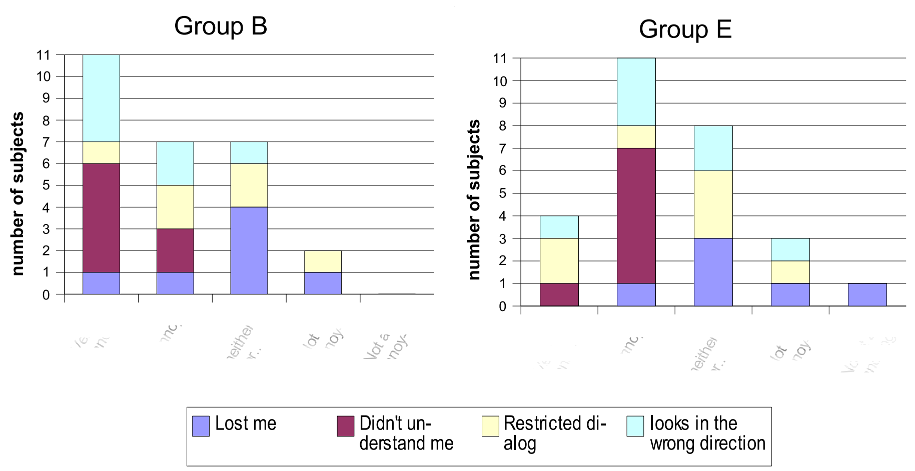


Figure 5.6.: The result of the question “How annoying were the following technical problems for you?”

	very annoying	annoying	neither... nor...	not annoying	not at all annoying
Group B	2.75	1.75	1.75	0.5	0
Group E	1	2.75	2	0.75	0.25

Table 5.2.: Average rating results of subjects concerning BIRON’s performance problems

On the issue of perceived interaction style, twice as many subjects in Group E agreed to the question as members of Group B (see Fig. 5.7).

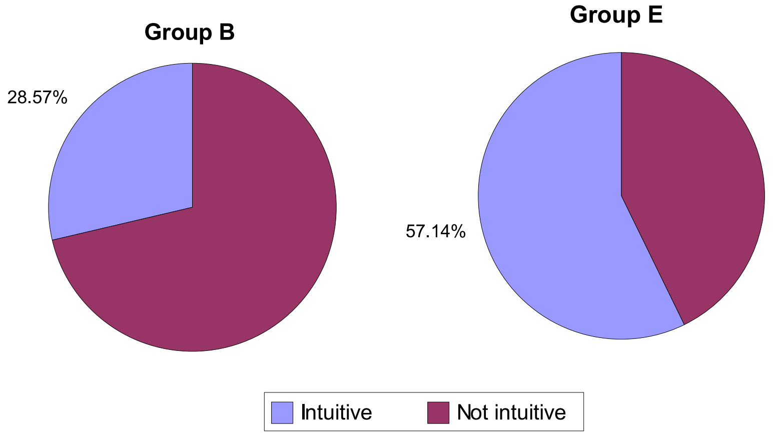


Figure 5.7.: The result of the question “Do you think the interaction with BIRON is intuitive?”

The chart in Fig. 5.8 illustrates the potential emotional effect of BIRON's different verbal behaviors. This result is a clear evidence for the effect: the overwhelming majority of Group E liked BIRON while only a small minority of Group B held the same emotion.

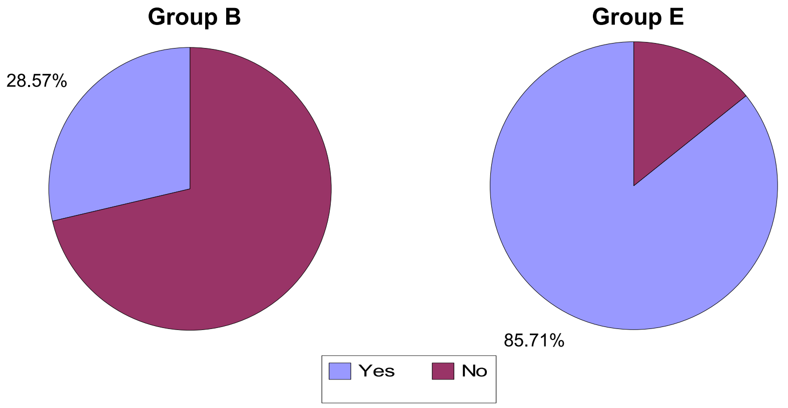


Figure 5.8.: The result of the question "Do you like BIRON?"

The results of this study suggest that the two types of BIRON were not only perceived as different, but the subjects' perception of BIRON's general performance and the interaction style were also affected. Furthermore, subjects of Group E (extrovert BIRON group) even felt emotionally animated in contrast to members of Group B. Therefore, the results of the study are a strong evidence for the hypothesis that the two social behaviors implemented for BIRON had positive impact on the subjective interaction quality, which was the initial goal for the implementation.

Observation and discussion

During the user study, observations were made from the perspective of the system developer [LW07]. In spite of the generally positive results of the study, the following three deficiencies of the system were identified, that seriously decreased the usability of the entire system:

1. **Insufficient robustness of the speech input control:** The major technical problem during the user study were speech recognition errors (as also can be seen in Fig. 5.6). The detailed performance analysis of the Speech Recognizer revealed that, beside its own problems, the control of the speech input by the Person Attention System (PAS) caused a lot of problems, too. Recall that the PAS only activated the Speech Recognizer if it successfully detected a face, sound and two legs in the same direction. Due to the unstructuredness and complexity of real environments, signals that the PAS received were often noisy which resulted in incorrect processing results. The consequence was that the entire speech processing sometimes could not be started because of processing errors in the PAS. The analysis of videos that were recorded during the experiment shows that, in the first run, only 63.47% of all the

utterances issued by the 14 subjects were forwarded by the PAS to the Speech Recognizer. In the second turn, it was slightly better: 74.91%. The reason for the improvement in the second run probably lay in the reduction of negative effects of user behaviors. By imitating the person in the demonstration video they behaved more “quiet” in speech, posture and gesture, which helped the robot system to produce correct results. Nevertheless, for most subjects, BIRON’s frequent silence was very confusing.

2. **The inability of the system to communicate BIRON’s perception and internal states:** In fact, even if BIRON did not react verbally to user’s speech input, the PAS did perceive differences in the current physical and social environment. The problem was just that BIRON had no possibility to communicate this perception. Consider the following case: One of the female subjects spoke with very low voice in high pitch so that the Speech Recognizer most of the time interpreted her voice as noise and did not forward it for further processing [1]. Puzzled by no reaction from the robot, the subject looked frequently in the direction of the experiment supervisor and asked why the robot did not react. Knowing that his own voice could also influence the robot’s perception of the environment, the supervisor tried to use gesture to make clear that he could not intervene. The subject seemed not to be able to interpret the meaning of the supervisor’s gesture and, therefore, went a few steps towards him so that she was out of the range where the robot could perceive her as a human. Then the subject came back and tried again, in vain. In this whole process, although the robot went through different internal states (person detected, person interaction intention recognized, person lost, person detected, person interaction intention recognized), there was no visible reaction from the robot. For the subject the interaction was a very frustrating experience. To communicate BIRON’s perception, its only output modality, speech, is obviously inappropriate. It is not possible, e.g., to let BIRON repeatedly generate output like “I see you, I see you, I can’t see you...” because such speech output would interrupt the “normal” interaction. However, this kind of information can be considerably better communicated with non-verbal modalities that are unobtrusive and can represent static information that is updated only occasionally [LW07].

3. **The lack of self-explanation:** BIRON operates mainly in the pattern of a finite state-machine, corresponding to the structure of its central module, the Execution Supervisor (see section 4.1.2). This means, BIRON can only execute one singe task at one moment. Furthermore, the “task” is defined in a purely technical sense. For example, “greeting” is a task because only if the user says “Hello” after the PAS detects her, the system is able to assign the user the status of “interaction partner” and focus its attention on her. However, for most users, “greeting” is hardly a task and it is often confusing why they could not show BIRON any object without saying “Hello”. Since the subjects were not aware of such pre-defined order of interaction state transitions, they often proposed illegal tasks, i.e., tasks that can not be executed in a given system state. In such situations, the IMS generated a Pseudo-Acceptance “I can’t do that right now.” However, this feedback turned

[1] The Speech Recognizer of BIRON is trained with predominately male voices and, therefore, does not work particularly well with female voices.

> out to be insufficient because the users still did not know what they could do so that BIRON could perform the task in the next moment. For such situations, the system should be able to behave cooperatively by explaining its capabilities given the specific system state and providing instant help.

The author's observation also revealed a problem in the experimental setup. Neither in the first nor in the second run the subjects were told the purpose of the interaction, i.e., showing the robot some objects so that it can use them to perform tasks later. The home tour scenario was obviously not an intuitive robot application field for many subjects: Five of the 14 subjects did not ask BIRON "What can you do?", as instructed in the first turn, but "What can you do *for me*?". Additionally, six of the subjects asked BIRON "What is this?" and pointed to an object after BIRON said "You can show me something and I can remember it." Apparently, some subjects had their own mental image of robots that they should do something for a human instead of vice versa. That also a robot needs to learn something through interaction with a human seems not be a part of the popular image of robots.

Based on the observations, the focus of the second IEC was put on the realization of functions and behaviors that increase the usability of the entire robot system.

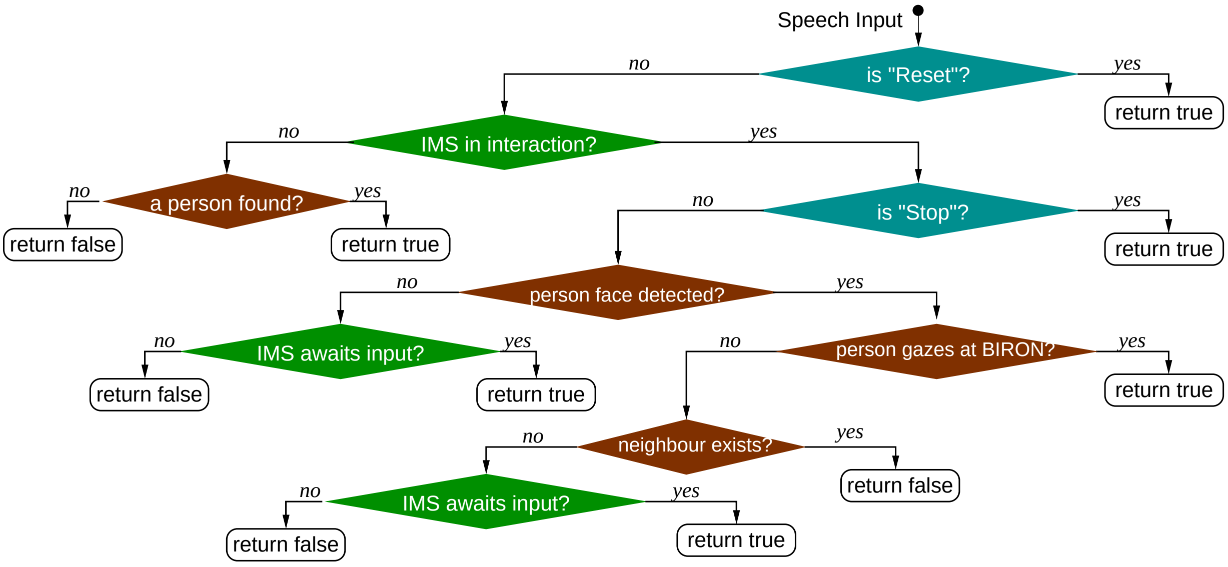
5.2. The second IEC: increasing usability

To solve the three usability problems listed above, in the second IEC, the control over the speech input was transfered from the PAS to the IMS, non-verbal feedback capabilities were realized and cooperative behaviors were implemented to help users out of tricky interaction situations.

5.2.1. Implementation

Controlling speech input

As presented in section 4.1.2 (page 78), the PAS recognizes user interaction intention by analyzing the combination of various percept: whether human legs are moving, whether the human is facing the robot and whether sound comes from the same direction as all the other signals. This approach is a purely bottom-up approach, i.e., the quality of a decision totally relies on the quality of signal processing. In a complex and unstructured environment as a human home, it is likely that the signal processing fails from time to time. If there are no other possibilities that can correct erroneous decisions of the PAS, the performance of the entire system can be seriously affected. To enable additional decision making mechanism, the IMS takes over the control of the speech input in the second IEC. More specifically, the Speech Recognizer (and the Speech



Decision based on speech input (◆), PAS information (◆), and the state of the IMS (◆).

Figure 5.9.: Decision making hierarchy: whether to consider the input from the Speech Recognizer (via Speech Understander)

Understander) are now active all the time and the IMS decides whether to consider their processing result or not. The goal is to draw upon other information to facilitate the decision making process.

In addition to the information delivered by the PAS (recall that there is a direct communication channel between the PAS and the IMS), also the semantic representation of the input and the state of the IMS, i.e., the current dialog context, are available to the IMS. This additional information enables the IMS to make the decision in a combined top-down, bottom-up manner. After extensive testing, the decision making hierarchy as illustrated in Fig. 5.9 was established. Two criteria guided the construction process of this hierarchy: practicability and potential costs of not considering certain input.

If the semantic representation of the input suggests a "reset", it will always be considered. This policy is established because it enables not only the user but also the developer to reset the entire system to its initial state at any time when something serious goes wrong. This command is especially handy to check whether the communication between system modules is still working, which is often the reason for "mysterious" technical problems. Of course the Speech Recognizer can have delivered the wrong result, however, the cost of not considering it would be much higher than considering it erroneously.

If the semantic representation of the input does not suggest a "reset", it is crucial to check out whether the IMS has already started an interaction with the user or not. If the IMS is not yet in interaction with a user, then it relies on the information of the PAS as to whether a person is detected. The IMS considers the speech input only if a person can be found.

If the IMS is in interaction with a user, the semantics of the input has to be looked at first. If it is a “stop”, it should be considered anyway. This policy has safety relevance because a robot should be able to be stopped in any situation to avoid possible damage to its user and environment.

If the input is semantically not a “stop”, the IMS draws upon information delivered by the PAS. If a human face can be detected and this face is oriented to the robot, then the speech should be considered. In fact, this is the most ideal case: a user is speaking to BIRON while looking at it. If no human face can be detected, the IMS looks at its own state: If the IMS expects user reply, i.e., if the IMS has just initiated an Exchange that needs to be explicitely grounded by the user, then the IMS considers the speech. If the IMS has no expectation, it abandons the input. This policy adds top-down knowledge into the decision making process and can remedy erroneous results of the signal processing by the PAS.

If a human face can be detected, but it is not oriented to the robot, then it is crucial to look at whether other persons are around. If so, then the current user is possibly talking to that person and the IMS dose not consider the input from the Speech Recognizer. If there are no other persons around, the IMS again checks out whether it expects user input and considers the speech only if the user is expected to reply.

This combined top-down, bottom-up decision making hierarchy turned out to be much more robust than the previous approach in the evaluation of the second IEC. In case that the IMS decides not to consider the processing result of the Speech Recognizer, which can still be an erroneous decision, users should be informed of what is happening. This was one of the reasons why non-verbal feedback capabilities were enabled in the IMS.

Enabling non-verbal feedback capabilities

The goal of the realization of non-verbal feedback capabilities is to enable BIRON to communicate (1) its perception, i.e., what it “sees” and “hears” and (2) its internal states during the interaction. Besides, non-verbal feedback concerning the social awareness of BIRON can prevent it from making too much social comments verbally and become annoying in a long interaction.

In the second IEC, non-verbal feedback is provided by a virtual character called “Mindi” (Fig. 5.10), which is displayed on the touch screen of BIRON. Mindi is a cartoon representation of BIRON and its large thought bubble is also visible. The content of the thought bubble is intended to communicate the perception of BIRON and the activities of Mindi essentially represent the system states. The choice of this character is motivated by Green and Severinson-Eklundh [GSE03], who advocate the powerfulness of a life-like character in HRI. In the following, how perception, internal states and social awareness of BIRON are represented by Mindi is discussed.

Regarding the perception, it is important to communicate why BIRON does not react in some situations, especially in case that the IMS decides not to process the incoming speech because of

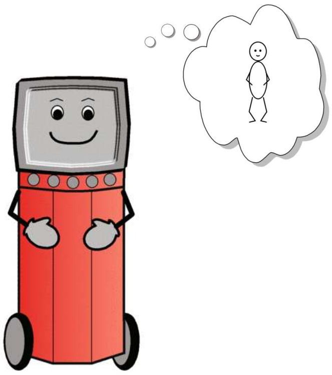


Figure 5.10.: Mindi in its default posture and its thought bubble (indicating that it can "see" its interaction partner)

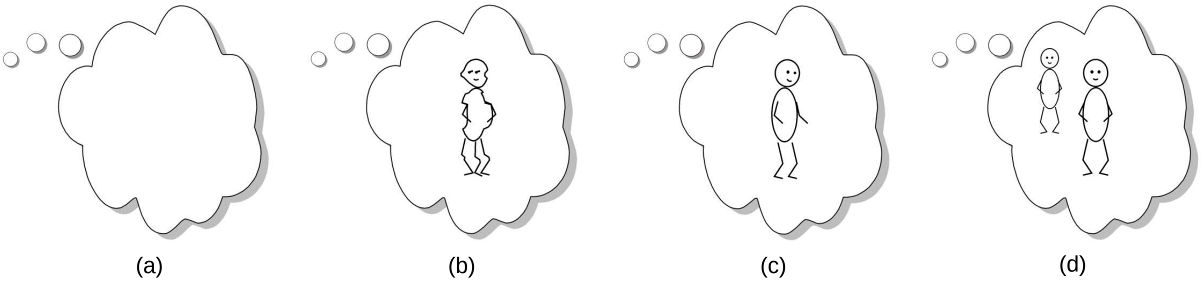


Figure 5.11.: Communicating BIRON's perception: (a) no person is found, (b) instable signals (no human face can be detected, animated), (c) the human does not gaze at BIRON, and (d) another person is visible (animated)

ambiguity of the signals. As discussed in the previous section (see Fig. 5.9), the reasons can be (1) no person is found; (2) no human face is detected; (3) the human does not gaze at BIRON; and (4) another person is visible. In these cases, different images are displayed in the thought bubble (see Figure 5.11).

The internal states of BIRON are represented with different postures of Mindi or the combination of Mindi and its thought bubble. Figure 5.12 shows three examples. Note, BIRON's internal states may not correspond to its visible behaviors. For example, when the state of BIRON is "following", it follows a person by trying to always keep a certain pre-defined distance to the user. This behavior enables BIRON to automatically adjust its speed to that of the user. When the user stops, BIRON also stops (because of the pre-defined distance to the user). This means, although BIRON is still in the state of "following", it has physically stopped. This inconsistency of an internal state and the external behavior is confusing for many users and they do not understand why they have to say "stop!" although the robot has already stopped. Therefore, for such cases, non-verbal feedback is enabled *additionally* to the verbal reply "OK, I'm following". Independently of the physical movement or "non-movement" of BIRON, as long as it is in the

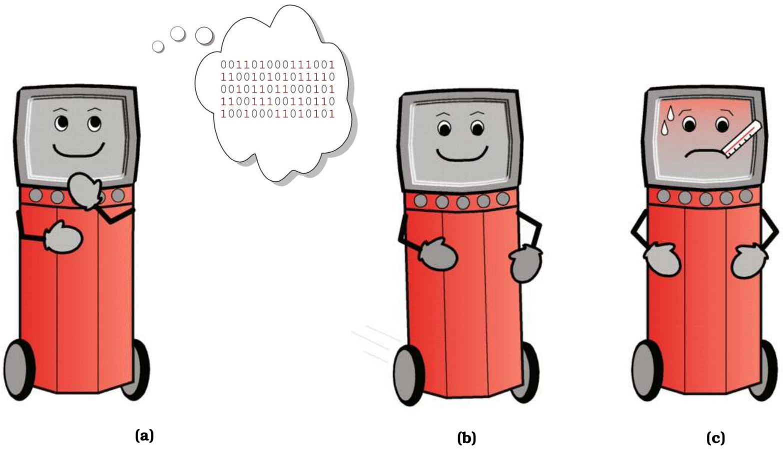


Figure 5.12.: Communicating BIRON's internal states: (a) BIRON is processing something (animated) (b) BIRON is following the user (animated), (c) BIRON "suffers" from severe technical failure.

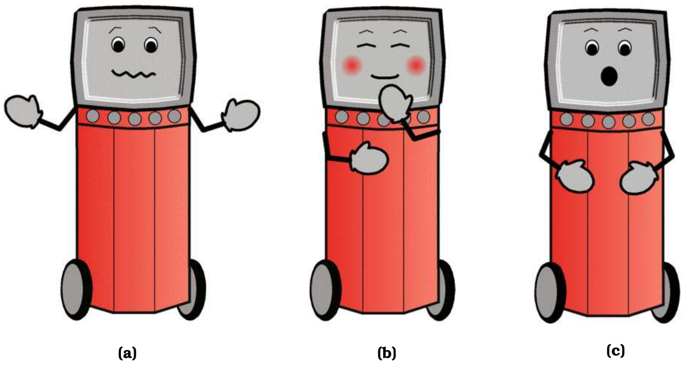
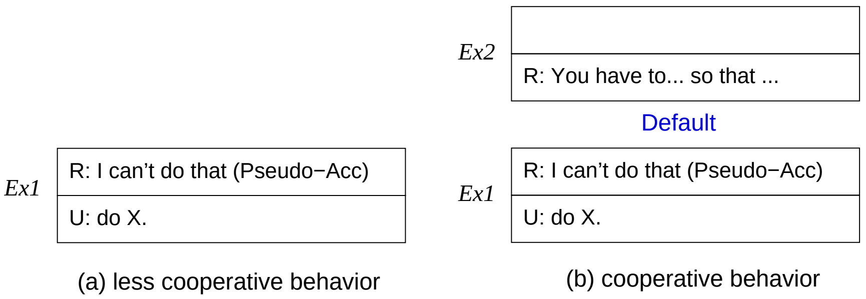


Figure 5.13.: Communicating BIRON's social awareness: (a) BIRON does not understand the user, (b) BIRON is embarrassed at its performance problem, (c) BIRON is surprised that the user suddenly leaves.

state "following", the animated image of Mindi is displayed (see Fig. 5.12 (b)). This is intended to make users aware of the actual internal state of BIRON.

The social awareness of BIRON, such as its awareness of its own performance problems (see section 5.1) is signaled by Mindi or verbal feedback or both, depending on the seriousness of the problem. Figure 5.13 shows some examples of how social awareness is represented by Mindi.

The activation of different images of Mindi and its thought bubble is controlled by the IMS. Whenever it is needed, the Dialog-, Robot- or UsabilityManager creates an IU, the BehaviorLayer of which is instantiated by the appropriate modality or modalities. More specifically, on the BehaviorLayer, the verbal generator can be instantiated with a plain text, which is intended to be synthesized by the Speech Synthesizer. It is also possible that the non-verbal generator


Ex1
R: I can't do that (Pseudo−Acc)
U: do X.
Ex2
R: You have to... so that ...
Default
Ex1
R: I can't do that (Pseudo−Acc)
U: do X.

(a) less cooperative behavior

(b) cooperative behavior

Figure 5.14.: Less cooperative vs. cooperative behaviors (U = User, R = Robot, Ex = Exchange, Acc = Acceptance)

is instantiated with the link to a specific set of images of Mindi and its thought bubble, which should be displayed on the touch screen. In some situations, both generators are instantiated to emphasize BIRON's motivation, e.g., in case of system state "following".

Non-verbal feedback capabilities enable the system to communicate more information than it is possible with only speech. However, this information merely represents system states as a *consequence* of the user's behavior, i.e., these capabilities can hardly influence the behavior of the user directly. For example, in case that a user proposes "illegal" commands that can not be executed by BIRON given the current system state, it is insufficient for Mindi to only demonstrate a sorrowful face. The system should be able to make the user aware of the problem and help her out of the situation. Such cooperative behaviors are also explicitly addressed in the second IEC and are discussed below.

Enabling cooperative behaviors

One of the advantages of the MMPDA model as an agent-based dialog model (compare to chapter 2) is that the interaction states are relatively independent of those of the domain tasks. This advantage enables the realization of different interaction behaviors given one single domain task state.

Consider the example of "illegal tasks": Once the user proposes such a task, the DialogManager of the IMS, as usual, represents the user's input as an IU initiating an Exchange, say Ex_1. Knowing that the task can not be executed in the current robot state, the DialogManager can create a "Pseudo-Acceptance", which only informs the user of the problem by generating "Sorry, I can't do that right now." (see Figure 5.14 (a)) Alternatively, the DialogManager can also choose to generate an additional Exchange, say Ex_2, to tell the user what she should do, given the current robot state, to achieve her goal (see Figure 5.14 (b)). The Ex_2 should be viewed as a Ew/oA (Exchange without Acceptance) because the user should have the possibility to either follow the IMS' suggestion or not.

The flexibility of the MMPDA model enables the realization of a number of cooperative behav-

Interaction situation	Cooperative behaviors by BIRON	Exchange initiated
User proposes a task that can not be executed given the current system state	initiating a confirmation question whether the user really meant it	a normal SupportEx
User confirms the illegal task	initiating a suggestion what she should do to achieve her goal	a Ew/oA with grounding relation Default
User proposes a task that can not be executed in general	initiating a confirmation question whether the user really meant it	a normal SupportEx
User confirms the impossible task	informing the user of general capabilities of BIRON	a Ew/oA with grounding relation Default
Interaction starts	informing the user of how to interpret Mindi	a (set of) normal DefaultEx
User proposes illegal tasks repeatedly or BIRON frequently has performance problems	initiating a self-reset	a normal DefaultEx
User agrees to reset BIRON	performing a self-reset	a Ew/oA with grounding relation Default

Table 5.3.: Cooperative behaviors and their realization (Ew/oA = Exchange without Acceptance)

iors, as summarized in Table 5.3. These behaviors and the interaction situations in which they should be exhibited are specified in an external configuration file so that they can be easily modified. In the IMS, the responsibility to realize these behaviors is taken by the UsabilityManager, which can be switched on and off when starting the IMS with different start parameters.

To find out whether these new functions and behaviors really helped to increase the usability of the system, a second user study was conducted to evaluate them.

5.2.2. Evaluation

In the evaluation, two versions of the system were contrasted to each other. The "cooperative BIRON" was able to demonstrate both social and cooperative behaviors (see section 5.1.1 on page 95 and section 5.2.1 on page 108), while the "basic BIRON" was passive and behaved in the same way as the basic BIRON in the previous user study. However, for both types of BIRON, the speech input control lay in the IMS and also Mindi was enabled for both cases.

Goal definition

The goal of the user study was to evaluate the effectiveness and the efficiency of the implemented behaviors in the second IEC. More specifically, it was intended to answer the following questions:

1. Is the control of the speech input by the IMS more robust than by the PAS?
2. Can the virtual character Mindi sufficiently convey information on BIRON's perception and internal states?
3. Are people who interacted with the cooperative BIRON more successful than those who interacted with the basic BIRON?

Method

For the study, eighteen subjects aged from 16 to 34 were recruited and the majority of them were in their mid-twenties. Before the interaction, all of them received a written instruction including the following information: the background of the home tour scenario, a brief description of BIRON's capabilities, the specification of their task in the experiment and a short list of utterances that BIRON understands. The subjects were given 5 minutes time to read the instruction before they interacted with BIRON. The main task in this experiment was to show BIRON a room. More specifically, the subjects were supposed to let BIRON follow them to the center of the room, tell BIRON that the room is the kitchen and lead BIRON back to their starting position (Fig. 5.15). It was not allowed for the subjects to take the instruction with them during the experiment. Nine of the subjects interacted with the cooperative BIRON (Group C) and the other nine with the basic BIRON (Group B). After the interaction, which took about 6 minutes in average, the subjects were asked to fill out a questionnaire.

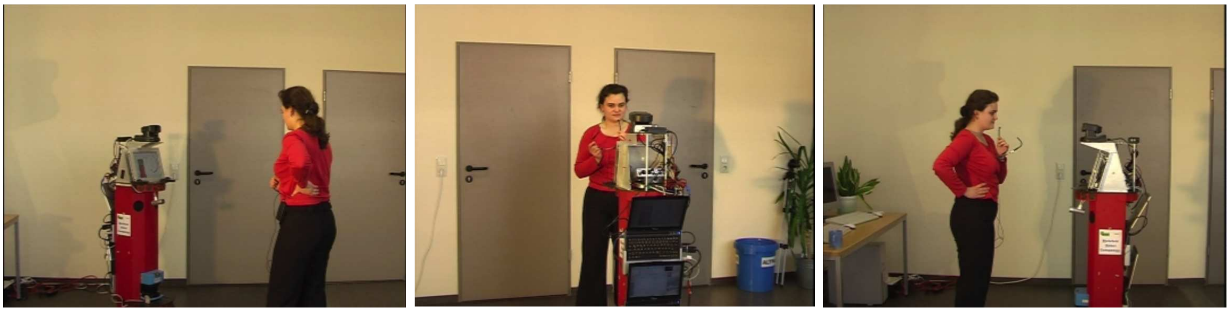


Figure 5.15.: The setup of the user study in the second IEC

To answer the first goal question, the reaction rate of BIRON was measured by analyzing the videos recorded during the experiment. The utterances issued by subjects in total and the frequency that BIRON actually generated a feedback *verbally* were counted. The result was intended to be compared to the observations of the first user study (see page 101). To answer the second goal question, the subjects were asked the question: "How often did you feel that you knew what was going on in the system?". The result of this question was also compared to that of the first study because subjects in that study were also asked this question. To answer the third goal question, a record about the interaction result (whether it was successful or not) as well as the length of the interaction was taken for each subject. The second and the third goal questions were intended to be answered separately. However, the results of the study revealed an unexpected but interesting relationship between these two questions.

Results

The results concerning BIRON reaction rate in the first and the second user study are contrasted in Table 5.4: the performance improvement in the second study is clearly visible. While in the two runs of the first user study about 1/3 and 1/4 of user utterances were ignored, the combined top-down, bottom-up approach to speech input control of the IMS archived a reaction rate of 96%. This result is more convincing given that BIRON was essentially immobile in the first user study and the environmental conditions were thus more stable than in the second study. Furthermore, even in the 4% of all the interaction situations where BIRON did not react verbally, non-verbal feedback generated by Mindi was visible to users that gave them hints as to what was happening. The usefulness of Mindi was further confirmed by the results of the second goal question.

Study	Interaction condition	User utterance total	Utterances BIRON reacted to	BIRON reaction rate
1st	1st run: interaction after a minimal instruction (no information on BIRON's abilities)	375	238	**63.47%**
	2nd run: interaction after a maximal instruction (demo-video and example dialog)	259	194	**74.90%**
2nd	interaction after a short instruction (home tour and language capabilities)	501	481	**96.00%**

Table 5.4.: BIRON's reaction rate in the first and the second user study

The results of the question "How often did you feel that you knew what was going on in the system?" in the two user studies are contrasted in Table 5.5. The improvement in the second user study is clear: While half of the subjects felt that they rarely knew what was happening in the first study, 72.22% in the second study believed that they knew it most of the time during the interaction.

	always	most of the time	sometimes	rarely	never
1st study	0.00%	28.57%	14.28%	**50.00%**	7.14%
2nd study	11.00%	**72.22%**	11.00%	0.00%	5.56%

Table 5.5.: User confidence in their knowledge about system states in the first and second user studies

The answer of the third goal question (whether members of the Group C were more successful than those of the Group B) was, at the first glance, negative. Members of both groups were similarly successful and they completed the task in similar time. Even their reaction to many questions from the questionnaire was similar: they seemed to be quite satisfied. However, closer examination of their behaviors revealed that, to achieve similar success as members of group C, group B members (1) more frequently made use of information provided by Mindi, and (2) had to be more active in the interaction. The following two paragraphs provide evidence for these two claims.

Figure 5.16 illustrates the result of the multiple choice question “What do you think about Mindi?” Although the majority of members of both groups agreed that they had fun with Mindi, five members of Group B believed that Mindi provided important information to them (comparing to only one person of Group C). Additionally, three from Group B thought that it is strange to see a small robot on a big robot, which was actually the major concern of the author when designing Mindi. In comparison to Group B members, Group C members seemed to be more or less indifferent to Mindi in general. The result of the question “How often did you look at Mindi during the interaction?” is shown in Fig. 5.17. While the majority of Group B members said that they did it most of the time, no clear tendency can be identified among Group C members. Apparently, members of Group B paid more attention to Mindi and appreciated it more than Group C members. The reason is probably that Group B members had to rely on information provided by Mindi more than Group C members.

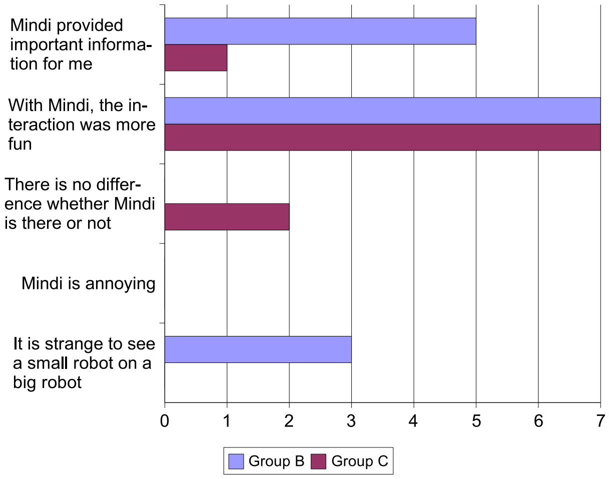


Figure 5.16.: The result of the question “What do you think about Mindi?”

The analysis of the videos that were recorded during the experiment revealed that members of Group B issued about 1/3 more utterances than Group C members although the average interaction lengths of both groups were similar (see Table 5.6). The reason for this discrepancy was that the cooperative BIRON often took initiative to help subjects (see page 109) so that the amount of utterances that they had to initiate (and also to repeat in case of illegal tasks) was lower than Group B members. This is an indication that the interaction between the cooperative BIRON and the Group C members was rather balanced, while Group B members had to be active all the time, e.g., to try different commands in case of illegal tasks.

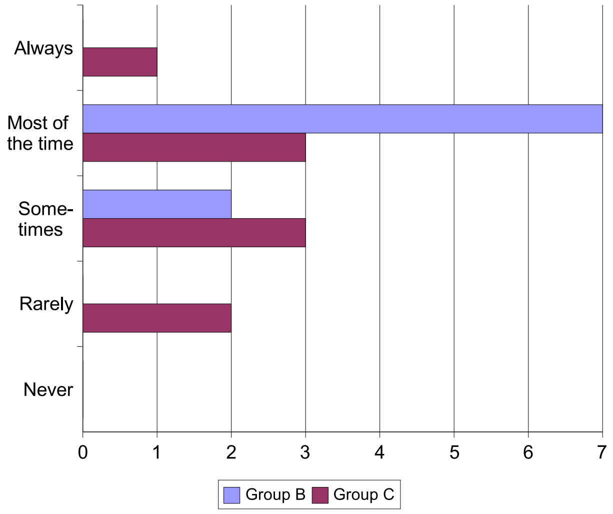


Figure 5.17.: The result of the multiple choice question “How often did you look at Mindi during the interaction?”

Group	User utterance total	Average interaction length
B	303	6.37 min.
C	198	6.26 min.

Table 5.6.: Total amount of user utterances vs. interaction length

As a summary, the performance improvement realized by the top-down, bottom-up approach to speech input control by the IMS was fully confirmed in the user study. Further, the ability of Mindi to communicate information on the perception and system states of BIRON was acknowledged by the majority of subjects. Although the similar success rate of members of both groups did not support author’s initial hypothesis, which was associated with the third goal question, the result did reveal a potentially higher cognitive load for Group B members. If the task that the subjects needed to accomplish was more complex, the increased cognitive load may have actually resulted in less success in task execution. This point is further discussed below.

Observation and discussion

The second user study was quite successful in general. Apart from two female subjects who spoke with low voice in high pitch and, therefore, had massive problems with the Speech Recognizer, all the other subjects successfully finished the task in reasonable time. They also seemed to be satisfied with the performance of BIRON. However, the observations made during the user study pose new challenge for the IMS.

The cooperative BIRON was quite verbose. It took both task-related and social initiatives to help and comfort its users. Since the subjects involved in this study interacted with BIRON only once and for the first time, this kind of behavior was welcome. However, if BIRON should "accompany" subjects on a long term basis, this behavior could become annoying to some subjects. The basic BIRON was not at all verbose and the subjects needed to be very committed to the interaction and to be attentive all the time. Although in the long term subjects may have better knowledge about how BIRON works, this kind of passive behavior of BIRON could create difficulty at the beginning. This means that users' different levels of interaction experience require different levels of initiative behaviors of BIRON.

In both the first and the second user study, various interaction styles of subjects were observed. Although all the subjects received the same instruction, some started the interaction with "How are you, robot?", some with "What can you do *for me*?" and some other strictly followed the instruction. This finding reveals that users' personal preferences in their interaction with a robot vary and it is conceivable that they hold different views on the same behavior of a robot.

As can be seen, personal differences in interaction experience and preference affect acceptability of a robot. Given that BIRON is intended to "live" with a human on a long term basis, it should be able to account for these differences. In another word, the IMS should be able to automatically adapt its choice of interactive behaviors to users. Although the realization of this ability is beyond the scope of the current work, it should be one of the most important goals in the future.

5.3. Summary

In this chapter, the implementation and the evaluation of various interactive functions and behaviors of the IMS were presented. These functions and behaviors included resolving multi-modal object references, exhibiting social awareness, controlling speech input, enabling non-verbal feedback capabilities and cooperative behaviors. The last three functions and behaviors were not planned at the beginning of the development, but based on the insight gained in the evaluation of existing ones. Nevertheless, these new functions and behaviors were easily implemented without any modifications of the MMPDA model itself, which confirms the powerfulness of the model in terms of its flexibility. The implemented functions and behaviors greatly contribute to the interaction quality of the robot system, as evident in the two evaluations. The observation made during the second user study suggests that the IMS should be able to adapt itself to various interaction experience and preference of users to account for long-term interaction. The realization of this ability is the focus of the future work, which will be further addressed in the next chapter.

6. Conclusions

The goal of the current work was to develop an interaction management system for a mobile robot companion. In comparison to many desktop computer applications, such a robot poses a number of new scientific questions that need to be addressed by its interaction management system. More specifically, the interaction management system should fulfill the following 8 requirements of Human-Robot Interaction (HRI): (1) handle cooperative interaction, (2) enable mixed-initiative interaction style, (3) separate interaction from domain task execution, (4) account for multi-modality of embodied interaction, (5) facilitate recognition of interaction initiated by users, (6) make use of different modalities in a meaningful way, (7) enable social behaviors, and (8) contribute to the usability of the entire robot system. In the current work, a powerful computational model of multi-modal grounding, the MMPDA model, was proposed that was implemented iteratively for the Interaction Management System (IMS) of the robot BIRON. This model and the implemented system completely fulfill the 8 requirements and, thus, stand out as one of the first comprehensive interaction models and systems in the field HRI.

The MMPDA model views embodied interaction as a cooperative process between interaction participants to establish mutual understanding. This process is called grounding. Grounding takes place in segments of interaction called Exchanges. They are stacked together via Grounding Relations and construct an interaction. An Exchange can be either initiated or grounded by a contribution of interaction partners in form of an Interaction Unit. Each interaction partner maintains her own private model of the on-going interaction which is updated upon arrival of a new Interaction Unit. The private model is thus organized as a stack containing ungrounded Exchanges. The states of the stack update following rules of a push-down automaton. This model fulfills the first four requirements of HRI:

Firstly, the MMPDA model views interaction as a cooperation between interaction partners, whether they are humans or artificial agents. The interaction partners are considered as possessing similar mental capabilities of interaction and their contributions to the interaction are, therefore, represented using the same structure: Interaction Units. These units update the private interaction model of interaction partners following the same rules. This way of viewing interaction naturally supports cooperative interaction that is required in learning scenarios.

Secondly, the concept grounding is based on the observation that interaction participants are willing to go on with the interaction only if they are sure that their contributions to the interaction are understood by their partners. When the partner does not provide the evidence of understanding, the initiator of a contribution is highly likely to initiate new contributions to support the grounding process of her interaction partner. This means, initiatives can be taken by either interaction participants whenever there are problems in understanding. Since the MMPDA model represents contributions of interaction partners using the same structure (the Interaction Units) either partner is allowed to take initiative by creating these units when it is needed. Thus, the MMPDA model account for mixed-initiative interaction style.

Thirdly, the state of the private interaction model of each interaction partner is only updated when an Exchange is initiated or grounded by an Interaction Unit. This means, the power that drives state transitions in the MMPDA model is not directly domain task states, but the grounding effects of individual Interaction Units. The association of domain task states with grounding effects of Interaction Units can be determined flexibly in the implementation and can also be updated online during the robot operation. This feature of the MMPDA model separates interaction from domain task execution as required by HRI.

Fourthly, the contributions of interaction partners are represented with Interaction Units that contain a modality-independent Motivation Layer and a modality-dependent Behavior Layer. On the Behavior Layer, verbal and non-verbal generators can be used to generate verbal messages and non-verbal expressions for the given motivation. This structure is able to separate the interaction motivations from their manifestation and thus account for multi-modality of interaction.

The implementation of the MMPDA model in the IMS for the robot BIRON went through two Implementation-Evaluation-Cycles (IEC), in which users played an important role in determination of implementation foci. In the course of the IECs, various functions and behaviors were realized that fulfill the last 4 requirements of HRI and their benefits were also proven in the two user studies:

The recognition of interaction that is initiated by users is a highly challenging task in unstructured real environments like a human home. More specifically, the system should be able to distinguish speech that is directed to the robot from human speech coming from other sound sources such as TVs and humans who are not involved in the interaction. The IMS recognizes the intended speech of BIRON's interaction partner by considering the semantic representation of the incoming speech, the current dialog context and the behavior of the interaction partner. In the evaluation, this combined top-down, bottom-up approach turned out to be much more robust than the purely bottom-up approach originally adopted by another system module of BIRON.

Based on the structure of Interaction Units, the IMS can easily handle different modalities for the input analysis and feedback generation. In the first IEC, this structure was used to facilitate the resolution of deictic gestures accompanying speech. During the interaction, a user utterance is represented as an Interaction Unit. More specifically, the verbal generator on the Behavior Layer of the Interaction Unit is instantiated with the utterance. If the utterance can not be fully understood because some deictic gestures seem to be involved, the IMS tries to detect them on the non-verbal generator (via other system modules of BIRON). If the motivation of the user's Interaction Unit can be identified this way, the IMS generates an Interaction Unit to ground the user's Unit. Otherwise, the IMS initiates clarification questions to resolve the issue. In the second IEC, non-verbal feedback capabilities were identified as crucial for usability reasons and were enabled with a virtual, life-like character, called Mindi. Mindi and its thought bubble are able to communicate information on BIRON's perception and internal states. This is realized by generating appropriate Interaction Units. For example, to communicate system perception, only the non-verbal generator of such Interaction Units is instantiated with a specific image of Mindi. To communicate internal states of the system, however, often both generators are instantiated because of the importance of such information. In contrast to subjects in the first user study, the majority of the subjects in the second user study agreed that they knew what was going on in the system most of the time.

The two social behaviors that were realized by the IMS exhibit BIRON's awareness of humans' presence and its own performance quality. Each time when a human is detected, the IMS generates an Interaction Unit and the Unit's verbal generator is instantiated with "Hello, human!". Users are then expected to ground this Interaction Unit. Further, the IMS measures the performance of BIRON by counting Exchanges that it has initiated to solve understanding problems, as manifested by relevant Grounding Relations of the Exchanges. Based on the performance, the IMS initiates Interaction Units to praise or to comfort users. In the evaluation, subjects who interacted with social aware BIRON appreciated BIRON more and tended to forgive its performance problems.

As to the last requirement of HRI, the combined top-down, bottom-up approach to speech input control and the realization of virtual character Mindi greatly increase the usability, as confirmed in the user study. Further, cooperative behaviors were also implemented to help users out of tricky interaction situations. More specifically, in the second IEC, whenever users propose illegal tasks the IMS generates additional Interaction Units to inform users what they should do to achieve their goals. If the user makes too many mistakes or BIRON repeatedly has performance problems, the IMS even initiates to reset itself to avoid interaction deadlocks. In the evaluation, such cooperative behaviors turned out to be helpful in reducing user's cognitive load during the interaction.

As can be seen, the MMPDA model is a highly powerful and flexible interaction model. It views interaction as cooperation, enables mixed-initiative interaction style, separates interaction states from domain task states and naturally handles multi-modality of embodied interaction. All these features of the model provide a solid basis for the implementation so that various effective functions and interactive behaviors were easily realized based on insight gained in user studies. Thus, the MMPDA model and the implemented IMS for the robot BIRON completely fulfill the 8 requirements of HRI and greatly contribute to the performance of the entire robot system.

The concept of Interaction Unit is further extendable to account for more sophisticated behavior generation, which can be done in the model as well as in the implementation. In the model, the relationship between verbal and non-verbal generators can be specified in a relatively general way so that a guideline for modality selection can be established. It is also possible that, based on different relationships between the two generators, different types of Interaction Units become necessary. In the implemented system a ModalityManager can be added, which selects, fuses and synchronizes the generation of multi-modal behaviors on the Behavior Layer.

To account for long-term interaction, adaptive behaviors should be realized. This means, decisions have to be made as to whether an Interaction Unit should be generated to provide users task-related hint or to comfort them. This decision making process should be based on observation of user interaction experience and preferences, e.g., how often did the user interact with BIRON, how often do they propose illegal tasks, whether they follow system's suggestions, whether they make social comments themselves, how do they react to BIRON's social comments and so on. These cues would enable the IMS to refine its behaviors and account for individual needs and preferences in long-term interaction.

> The dream robot of the author is one that is intelligent and human, and of course can perfectly perform the home tour scenario...

Bibliography

[ABD$^+$01] J. Allen, D. Byron, M. Dzikovska, G. Ferguson, L. Galescu, and A. Stent. Towards conversational human-computer interaction. *AI Magazine*, 22(4), 2001.

[ACC$^+$94] G. Antoniol, B. Caprile, A. Cimatti, R. Fiutem, and G. Lazzari. Experiencing real-life interaction with the experimental platform of maia. In *Proc. 1st European Workshop on Human Comfort and Security*, 1994.

[ADB04] G. Abowd A. Dix, J. Finlay and R. Beale. *Human Computer Interaction*. Prentice Hall, 3 edition, 2004.

[AFS$^+$05] J. Allen, G. Ferguson, M. Swift, A. Stent, S. Stoness, L. Galescu, N. Chambers, E. Campana, and G. Aist. Two diverse systems built using generic components for spoken dialogue (recent progress on trips). In *Proc. 43rd Annual Meeting of the Association for Computational Linguistics*, 2005.

[Aib] Aibo. http://www.sony.net/Products/aibo/.

[All95] J. Allen. *Natural Language Processing*. Benjamin Cummings Publishing Company, 2 edition, 1995.

[All01] J. Allwood. Cooperation and flexibility in multimodal communication. In H. Bunt and R. J. Beun, editors, *Cooperative Multimodal Communication*, Lecture Notes in Computer Science 2155. Springer Verlag, 2001.

[ANA92] J. Allwood, J. Nivre, and E. Ahlsén. On the semantics and pragmatics of linguistic feedback. *Journal of Semantics*, 9(1):1–26, 1992.

[AO95] H. Aust and M. Oerder. Dialogue control in automatic inquiry systems. In *Proc. ESCA Workshop on Spoken Dialogue Systems*, 1995.

[AOSS95] H. Aust, M. Oerder, F. Seide, and V. Steinbiss. The philips automatic train timetable information system. *Speech Communication*, 17, 1995.

[AS91] J. Allen and L. Schubert. The trains project. Technical report, Computer Science Dept, Universtiy of Rochester, 1991.

[AS05] K. Aoyama and H. Shimomura. Real world speech interaction with a humanoid robot on a layered robot behavior control architecture. In *Proc. Int. Conf. on Robotics and Automation*, 2005.

[BC01] T. Bickmore and J. Cassell. Relational agents: a model and implementation of building user trust. In *Proc. Int. Conf. on Human Factors in Computing Systems*, 2001.

[BCF$^+$98] W. Burgard, A. B. Cremers, D. Fox, D. Hahnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. The interactive museum tour-guide robot. In *Proc. 15th Nat. Conf. on Artificial Intelligence*, 1998.

[BFJ$^+$05] M. Bennewitz, F. Faber, D. Joho, M. Schreiber, and Sven Behnke. Multimodal conversation between a humanoid robot and multiple persons. In *Proc. Workshop on Modular Construction of Human-Like Intelligence, the Twentieth nat. Conf. on Artificial Intelligence*, 2005.

[BG02] R. Bischoff and V. Graefe. Dependable multimodal communication and interaction with robotic assistants. In *Proc. Int. Workshop on Robot-Human Interactive Communication (ROMAN)*, 2002.

[BH95] S. E. Brennan and E. Hulteen. Interaction and feedback in a spoken language system: A theoretical framework. *Knowledge-based Systems*, 8(2–3):143–151, 1995.

[BMP$^+$00] C. Benoit, J.-C. Martin, C. Pelachaud, L. Schomaker, and B. Suhm. Audio-visual and multimodal speechbased systems. In D. Gibbon, I. Mertins, and R. Moore, editors, *Handbook of Multimodal and Spoken Dialogue Systems: Resources, Terminology and Product Evaluation*. Kluwer, 2000.

[Bol80] R. A. Bolt. Put-that-there: Voice and gesture at the graphics interface. *Computer Graphics*, 14, 1980.

[Bre00] C. Breazeal. *Sociable Machines: Expressive Social Exchange Between Humans and Robots*. dissertation, Department of Electrical Engineering and Computer Science, MIT, 2000.

[Bro86] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1), 1986.

[Bro89] A. R. Brooks. A robot that walks; emergent behaviors from a carefully evolved network. MIT AI Lab Memo 1091, 1989.

[BWK$^+$03] H.-J. Böhme, T. Wilhelm, J. Key, C. Schauer, C. Schröter, H.-M. Groß, and T. Hempel. An approach to multi-modal human-machine interaction for intelligent service robots. *Robotics and Autonomous Systems*, 2003.

[Cah92] Janet E. Cahn. A computational architecture for the progression of mutual understanding in dialog. Technical report, Music and Cognition Group, Media Laboratory. Massachusetts Institute of Technology, 1992.

[Car92] R. Carpenter. *The logic of typed feature structures*. Cambridge University Press., 1992.

[Cas00] J. Cassell. More than just another pretty face: Embodied conversational interface agents. *Communications of the ACM*, 43(4), 2000.

[CB99] J. E. Cahn and S. E. Brennan. A psychological model of grounding and repair in dialog. In *Proc. Fall 1999 AAAI Symp. on Psychological Models of Communication in Collaborative Systems*, 1999.

[CBCV00] J. Cassell, T. Bickmore, L. Campbell, and H. Vilhjalmsson. Human conversation as a system framework: Designing embodied conversational agents. In J. Cassell, J. Sullivan, S. Prevost, and E. Churchill, editors, *Embodied conversational agents*. MIT Press, 2000.

[CC99] J. Chu-Carroll. Form-based reasoning for mixed-initiative dialogue management in information-query systems. In *Proc. European Conference on Speech Communication and Technology*, 1999.

[CCD00] A. Colburn, M. F. Cohen, and S. M. Drucker. The role of eye gaze in avatar mediated conversational interfaces. Technical Report MSR-TR-2000-81, Microsoft research, 2000.

[CG04] A. Cappelli and E. Giovannetti. Human-robot interaction. *Intelligenza Artificiale*, 1(2), 2004.

[CL99] R. Cooper and S. Larsson. Dialogue moves and information states. In *Proc. 3rd Int. workshop on computational linguistics*, 1999.

[Cla92] H. H. Clark, editor. *Arenas of Language Use*. University of Chicago Press, 1992.

[CMN86] S. K. Card, T. P. Moran, and A. Newell. The model human processor: An engineering model of human performance. In K. K. L. Boff and J. Thomas, editors, *Handbook of Perception and Human Performance*. New York: John Wiley and Sons, 1986.

[CO71] W. S. Condon and W. D. Osgton. Speech and bodymotion synchrony of the speaker-hearer. In D. H. Hortonand J. J. Jenkins, editor, *The perception of Language*. 1971.

[Cog04] Cogniron. http://www.cogniron.org/, 2004.

[CPB$^+$94] J. Cassell, C. Pelachaud, N. Badler, M. Steedman, B. Achorn, T. Becket, B. Douville, S. Prevost, and M. Stone. Animated converstaion: Rule-based generation of facial expression, gesture and spoken intonation for multiple conversational agents. In *Proc. 21st Annual Conf. on Computer Graphics and Interactive Techniques*, 1994.

[CSB$^+$02] J. Cassell, T. Stocky, T. Bickmore, Y. Gao, Y. Nakano, K. Ryokai, D. Tversky, C. Vaucelle, and H H. Vilhjalmsson. Mack: Media lab autonomous conversational kiosk. In *Proc. of Imagina02*, 2002.

[CT99] J. Cassell and K. Thórisson. The power of a nod and a glance: envelope vs. emotional feedback in animated conversational agents. *Applied Artificial Intelligence*, 13, 1999.

[DBD98] L. Dybkj, N.O. Bernsen, and H. Dybkj. A methodology for diagnostic evaluation of spoken human-machine interaction. *Int. Journal of Human-Computer studies*, 1998.

[DG95] M. Danieli and E. Gerbino. Metrics for evaluating dialogue strategies in a spoken language system. In *Working Notes of the AAAI Spring Symp. on Empirical Methods on Discourse Interpretation and Generation*, pages 34–39. AAAI, 1995.

[DJ00] B. R. Duffy and G. Joue. Intelligent robots: The question of embodiment. In *Proc. BRAIN-MACHINE*, Ankara, 2000.

[DTS96] P. Dillenbourg, D. Traum, and D. Schneider. Grounding in multi-modal task-oriented collaboration. In *Proc. EuroAI & Education Conference*, 1996.

[Dun72] S. D. Duncan. Some signals and rules for taking speaking turns in conversations. *Journal of personality and social psychology*, 23, 1972.

[DWK$^+$05] K. Dautenhahn, S. Woods, C. Kaouri, M. Walters, K. Koay, and I. Werry. What is a robot companion - friend, assistant or butler? In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2005.

[EE75] H. Eysenck and S. Eysenck. *Manual of the Eysenck personality questionnaire*. Hodder and Stoughton, 1975.

[FA98] G. Ferguson and J. Allen. Trips: An integrated intelligent problem-solving assistant. In *Proc. Nat. Conf. on Artificial Intelligence*, 1998.

[FAM96] G. Ferguson, J Allen, and B. Miller. Trains-95: Towards a mixed-initiative planning assistant. In *Proc. Int. Conf. on AI Planning Systems*, 1996.

[FAM98] J. Fry, H. Asoh, and T. Matsui. Natural dialogue with the jijo-2 office robot. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1998.

[Fes] Festival. http://www.cstr.ed.ac.uk/projects/festival/.

[Fin95] G. A. Fink. *Integration von Spracherkennung und Sprachverstehen. Dissertationen zur Künstlichen Intelligenz*. Dissertation, Bielefeld University, Faculty of Technology, 1995.

[FKH$^+$05] J. Fritsch, M. Kleinehagenbrock, A. Haasch, S. Wrede, and G. Sagerer. A flexible infrastructure for the development of a robot companion with extensible HRI-capabilities. In *Proc. IEEE Int. Conf. on Robotics and Automation*, Barcelona, Spain, 2005.

[FKL$^+$04] J. Fritsch, M. Kleinehagenbrock, S. Lang, G. A. Fink, and G. Sagerer. Audiovisual person tracking with a mobile robot. In *Proc. Int. Conf. on Intelligent Autonomous Systems*, 2004.

[GE01] A. Green and K. S. Eklundh. Task-oriented dialogue for cero: a user-centered approach. In *Proc.10th IEEE Int. Workshop on Robot and Human Interactive Communication*, 2001.

[GLD99] E. Guglielmelli, C. Laschi, and P. Dario. Robots for personal use: Humanoids vs. distributed systems. In *Proc. Int. Symp. on Humanoid Robots (HURO)*, 1999.

[GMP$^+$96] D. Goddeau, H. Meng, J. Polifroni, S. Seneff, and S. Busayapongchai. A form-based dialog manager for spoken language applications. In *Proc. International Conference on Spoken Language Processing*, 1996.

[Gri75] P. Grice. Logic and conversation. In *Speech Acts, Syntax and Semantics III*. Academic Press, 1975.

[GS86] B. Grosz and C. Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(175–204), 1986.

[GSE03] A. Green and K. Severinson-Eklundh. Designing for learnability in human-robot communication. *IEEE Transactions on Industrial Electronics*, 50(4):644–650, 2003.

[Haa07] A. Haasch. *Attention-controlled Acquisition of a Qualitative Scene Model for Mobile Robots*. Dissertation, Bielefeld University, Faculty of Technology, 2007.

[HDM$^+$90] L. Hirschman, A. D. Dahl, D. P. McKay, L. M. Norton, and M. C. Linebarger. Beyond class a: A proposal for automatic evaluation of discourse. In *Proc. Speech and Natural Language Workshop*, 1990.

[Heg79] T. G. Hegstrom. Message impact: What percentage is nonverbal? *Western journal of speech communication*, 1979.

[HHFS05] A. Haasch, N. Hofemann, J. Fritsch, and G. Sagerer. A multi-modal object attention system for a mobile robot. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2005.

[Hor93] I. Horswill. Polly: A vision-based artificial agent. In *Proc. 11th Nat. Conf. on Artificial Intelligence*, 1993.

[Hor96] I. Horswill. The design of the polly system. Technical report, Northwestern University, 1996.

[HTHS04] L. M. Hiatt, J. G. Trafton, A. M. Harrison, and A. C. Schultz. A cognitive model of spatial perspective taking. In *Proc. Int. Conf. of Cognitive Modeling*, 2004.

[Hue07a] H. Huettenrauch. *From HCI to HRI: Designing Interaction for a Service Robot*. Dissertation, Royal Institute of Technology, Stockholm, 2007.

[Hue07b] S. Huewel. *Robustes Verstehen gesprochener Sprache in einem multimodalen Roboter-Szenario*. Dissertation, Bielefeld University, Faculty of Technology, 2007.

[Hul99] J. Hulstijn. Modelling usability: development methods for dialogue systems. In *Proc. IJCAI99 Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, 1999.

[HW06] S. Hüwel and B. Wrede. Situated speech understanding for robust multi-modal human-robot communication. In *Proc. COLING/ACL 2006 Main Conference Poster Sessions*, 2006.

[HWS06] S. Hüwel, B. Wrede, and G. Sagerer. Robust speech understanding for multi-modal human-robot communication. In *Proc. 15th Int. Symp. on Robot and Human Interactive Communication*, 2006.

[ICLC99] J. M. Iverson, O. Capirci, E. Longobardi, and M. C. Caselli. Gesturing in mother-child interactions. *Cognitive Develpment*, 14(1):57–75, 1999.

[JL02] S. E. Jones and C. D. LeBaron. Research on the relationship between verbal and nonverbal communication: Emerging integrations. *Journal of Communication*, 52(3), 2002.

[Kap00] F. Kaplan. Talking AIBO: First experimentation of verbal interactions with an autonomous four-legged robot. In *Proc. of the CELE-Twente Workshop on Interacting Agents*, Workshop series on (Human-)Agent Interaction and Agent Learning, 2000.

[KCH+04] G. Kim, W. Chung, S. Han, K. Kim, M. Kim, and R. H. Shinn. The autonomous tour-guide robot jinny. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2004.

[KFS04] M. Kleinehagenbrock, J. Fritsch, and G. Sagerer. Supporting advanced interaction capabilities on a mobile robot with a flexible control system. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Sendai, Japan, 2004.

[Kha98] Z. Khan. Attitudes towards intelligent service robots. Technical Report TRITA-NA-P9821, IpLab, NADA, Royal Institute of Technology, Sweden, 1998.

[Kle05] M. Kleinehagenbrock. *Interaktive Verhaltenssteuerung für Robot Companions*. Dissertation, Bielefeld University, Faculty of Technology, 2005.

[KM96] T. Koda and P. Maes. Agents with faces: The effects of personification of agents. In *Proc. Int. Conf on Human Computer Interaction*, 1996.

[KP75] L. Karttunen and S. Peters. Conventional implicature of montague grammer. In *Proc. 1st Annual Meeting of the Berkely Linguistic Society, University of California*, 1975.

[KPC$^+$03] B. J. A. Krose, J. M. Porta, K. Crucq, A. J. N. van Breemen, M. Nuttin, and E. Demeester. Lino, the user-interface robot. In *Proc. 1st Euro. Symp. on Ambience Intelligence*, 2003.

[LA87] D. J. Litman and J. Allen. A plan recognition model for subdialogues in conversation. *Cognitive Science*, 1987.

[Lan05] S. Lang. *Multimodale Aufmerksamkeitssteuerung fuer einen mobilen Roboter*. Dissertation, Universitaet Bielefeld, Technische Fakultaet, 2005.

[LB94] L. Larsen and A. Baeekgaard. Rapid prototyping of a dialogue system using a generic dialogue development platform. In *Proc. International Conference on Spoken Language Processing*, 1994.

[LCK$^+$97] J. Lester, S. Converse, S. Kahler, S. Barlow, S. Stone, and R. Bhogal. The persona effect: affective impact of animated pedagogical agents. In *Proc. Int. Conf. on Human Factors in Computing Systems*, 1997.

[LHW$^+$05] S. Li, A. Haasch, B. Wrede, J. Fritsch, and G. Sagerer. Human-style interaction with a robot for cooperative learning of scene objects. In *Proc. Int. Conf. on Multimodal Interfaces*, 2005.

[LKF$^+$04] S. Li, M. Kleinehagenbrock, J. Fritsch, B. Wrede, and G. Sagerer. “BIRON, let me show you something”: Evaluating the interaction with a robot companion. In *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics, Special Session on Human-Robot Interaction*, 2004.

[LKH$^+$03] S. Lang, M. Kleinehagenbrock, S. Hohenner, J. Fritsch, A. Fink, and G. Sagerer. Providing the basis for human-robot-interaction: A multi-modal attention system for a mobile robot. In *Proc. Int. Conf. on Multimodal Interfaces*, 2003.

[LP99] D. J. Litman and S. Pan. Empirically evaluating an adaptable spoken dialogue system. In *Proc. 7th Int. Conf. on User Modelling*, 1999.

[LT00] S. Larsson and D. Traum. Information state and dialogue management in the trindi dialogue move engine toolkit. *Natrual Language Engineering*, 6(3–4), 2000.

[LW07] S. Li and B. Wrede. Why and how to model multi-modal interaction for a mobile robot companion. In *Proc. AAAI Spring Symp. on Interaction Challenges for Intelligent Assistants*, Stanford, 2007.

[LWS06] S. Li, B. Wrede, and G. Sagerer. A computational model of multi-modal grounding. In *Proc. SIGdial workshop on discourse and dialog, in conjunction with COLING/ACL*, Sydney, Australia, July 2006. ACL Press.

[MAF$^+$99] T. Matsui, H. Asoh, J. Fry, Y. Motomura, F. Asano, T. Kurita, I. Hara, and N. Otsu. Integrated natural spoken dialogue system of jijo-2 mobile robot for office services,. In *Proc. AAAI National Conference and Innovative Applications of Artificial Intelligence Conference*, 1999.

[MAMJ01] H. McBreen, J. Anderson, and M M. Jack. Evaluating 3d embodied conversational agents in contrasting vrml retail applications. In *Proc. Workshop Multimodal communication and context in embodied agents*, 2001.

[McT02] M. F. McTear. Spoken dialogue technology: enabling the conversational interface. *ACM Computing Surveys*, 34(1):90–169, 2002.

[McT04] M. F. McTear. *Spoken Dialogue Technology: Toward the Conversational User Interface*. Springer Verlag, 2004.

[MhA$^+$00] T. Matsui, A. hideki, F. Asano, T. Kurita, I. Hara, Y. Motomura, K. Itou, and J. Fry. Integration of real-world interaction functions on the jijo-2 office robot. In *Proc. Real World Computing Symp.*, 2000.

[MPR$^+$02] M. Montemerlo, J. Pineau, N. Roy, S. Thrun, and V. Verma. Experiences with a mobile robotic guide for the elderly. In *Proc. Nat. Conf. on Artificial Intelligence (AAAI)*, 2002.

[MSJW00] H. M. McBreen, P. Shade, M. A. Jack, and P. J. Wyard. Experimental assessment of the effectiveness of synthetic personae for multi-modal e-retail applications. In *Proc. 4th Int. Conf. on Autonomous Agents*, 2000.

[Nag04] Y. Nagai. *Understanding the Development of Joint Attention from a Viewpoint of Cognitive Developmental Robotics*. PhD thesis, Osaka University, 2004.

[NBG$^+$99] I. Nourbakhsh, J. Bobenage, S. Grange, R. Lutz, R. Meyer, and A. Soto. An affective mobile educator with a full-time job. *Artificial Intelligence*, 114(1–2), 1999.

[Nil84] N. J. Nilsson. Shakey the robot. Technical Note 323, AI Center, SRI International, Menlo Park, CA, 1984.

[NIL00] C. Nass, K. Isbister, and E.-J. Lee. Truth is beauty: Researching embodied conversational agents. In J. Cassell, J. Sullivan, S. Prevost, and E. Churchill, editors, *Embodied Conversational Agents*. MIT Press, 2000.

[NRSC03] Y. I. Nakano, G. Reinstein, T. Stocky, and J. Cassell. Towards a model of face-to-face grounding. In *Proc. Annual Meeting of the Association for Computational Linguistics*, 2003.

[NT99] C. Nakatani and D. Traum. Coding discourse structure in dialogue (version 1.0). Technical Report UMIACS-TR-99-03, University of Maryland, 1999.

[OC00] S. L. Oviatt and P. R. Cohen. Multimodal systems that process what comes naturally. *Communications of the ACM*, 43(3), 2000.

[OCW$^+$00] S. L. Oviatt, P. R. Cohen, L. Wu, J. Vergo, L. Duncan, B. Suhm, J. Bers, T. Holzman, T. Winograd, J. Landay, J. Larson, and D. Ferro. Designing the user interface for multimodal speech and pen-based gesture applications: State-of-the-art systems and future research directions. *Human Computer Interaction*, 15(4):263–322, 2000.

[Ore83] B. Oreström. Turn-taking in english conversation. In *Lund Studies in English*, number 68. Gleerup, 1983.

[Ovi03] S. L. Oviatt. Multimodal interfaces. In J. JACKO and A. SEARS, editors, *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*. Lawrence Erlbaum Assoc, 2003.

[PAB03] N. Pfleger, J. Alexandersson, and T. Becker. A robust and generic discourse model for multimodal dialogue. In *Proc. 3rd Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, 2003.

[Pec93] J. Peckham. A new generation of spoken dialogue systems: Results and lessons from the sundial project. In *Proc. 3rd Euro. Conf. on Speech Communication and Technology*, 1993.

[PH98] V. Pavlovic and T. S. Huang. Multimodal prediction and classification on audio-visual features. In *Proc. AAAI Workshop on Representations for Multi-modal Human-Computer Interaction*, 1998.

[PRBO96] J. Potjer, A. Russel, L. Boves, and E. D. Os. Subjective and objective evaluation of two types of dialogues in a call assistance service. In *Proc. IEEE Third Workshop Interactive Voice Technology for Telecommunications Applications*, 1996.

[Qri] Qrio. http://www.sony.net/SonyInfo/QRIO/.

[RBF$^+$00] N. Roy, G. Baltus, D. Fox, F. Gemperle, J. Goetz, T. Hirsch, D. Magaritis, M. Montemerlo, J. Pineau, J. Schulte, and S. Thrun. Towards personal service robots for the elderly. In *Proc. Int. Workshop on Interactive Robotics and Entertainment*, 2000.

[RDN02] Z. Ruttkay, C. Dormann, and H. Noot. Evaluating ecas - what and how? In *Proc. Embodied conversational agents - let's specify and evaluate them!*, 2002.

[RJ00] J. Rickel and W.L. Johnson. Task-oriented collaboration with embodied agents in virtual world. In J. Cassell, J. Sullivan, S. Prevost, and E. Churchill, editors, *Embodied conversational agents*. MIT Press, 2000.

[RS97] C. Rich and C. L. Sidner. Collagen: When agents collaborate with people. In *Proc. Int. Conf. on Autonomous Agents*, 1997.

[Sad94] M. D. Sadek. Communication theory = rationality principles + communicative act models. In *Proc. AAAI 94 workshop on modeling of interagent communication*, 1994.

[SAS01] D. Spiliotopoulos, I. Androutsopoulos, and C.D. Spyropoulos. Human-robot interaction based on spoken natural language dialogue. In *Proc. Euro. Workshop. Service and Humanoid Robots*, 2001.

[SB05] H. Shi and J. Bateman. Developing human-robot dialogue management formally. In *Proc. Symp. on Dialogue Modelling and Generation*, 2005.

[SBG+03] R. Simmons, A. Bruce, D. Goldberg, A. Goode, M. Montemerlo, N. Roy, B. Sellner, C. Urmson, A. Schultz, W. Adams, M. Bugajska, M. MacMahon, J. Mink, D. Perzanowski, S. Rosenthal, S. Thomas, I. Horswill, R. Zubek, D. Kortenkamp, B. Wolfe, T. Milam, and B. Maxwell. Grace and george: Autonomous robots for the aaai robot challenge. In W. Smart and M. Bugajska, editors, *AAAI Mobile Robot Competition 2003 : Papers from the AAAI Workshop*. AAAI Press, 2003.

[SdOB99] J. Sturm, E. den Os, and L. Boves. Dialogue management in the dutch arise train timetable information system. In *Proc. European Conference on Speech Communication and Technology*, 1999.

[SF93] A. Simpson and N. A. Fraser. Black box and glass box evaluation of the sundial system. In *Proc. 3rd Euro. Conf. on Speech Communication and Technology*, 1993.

[SFC+96] M. D. Sadek, A. Ferrieux, A. Cozannet, P. Bretier, F. Panaget, and J. Simonin. Effective human-computer cooperative spoken dialogue: The ags demonstrator. In *Proc. Int. Conf. on Spoken Language Processing*, 1996.

[SGH+04] W. Swartout, J. Gratch, R. Hill, E. Hovy, S. Marsella, J. Rickel, and D. Traum. Toward virtual humans. In *Working notes of the AAAI Fall Symp. on Achieving Human-Level Intelligence through Integrated Systems and Research*, 2004.

[SH94] R. Smith and D. R. Hipp. *Spoken natural language dialogue systems: A practical approach*. Oxford University Press, 1994.

[Sip97] Michael Sipser. *Introduction to the Theory of Computation*. PWS Publishing, 1997.

[SKLL04] C. L. Sidner, C. Kidd, C. Lee, and N Lesh. Where to look: A study of human-robot engagement. In *Proc. Intelligent User Interfaces*, 2004.

[SLW+07] T. Spexard, S. Li, B. Wrede, M. Hanheide, E. Topp, and H. Huettenrauch. Interaction awareness for joint environment exploration. In *Proc. Int. Symp. on Robot and Human Interactive Communication, Special Session: Situation Awareness in Social Robots*, 2007.

[SP00] S. Seneff and J. Polifroni. Dialogue management in the mercury flight reservation system. In *Proc. ANLP/NAACL Workshop on Conversational Systems*, 2000.

[SS73] E. A. Schegloff and H. Sacks. Opening up closings. *Semiotica*, pages 289–327, 1973.

[Sta78] R. C. Stalnaker. Assertion. In P. Cole, editor, *Syntax and semantics*, volume 9, pages 315–332. Academic, New York, 1978.

[Ste84] A. B. Stenström. Questions and responses in english conversation. In *Lund Studies in English*, number 68. Gleerup, 1984.

[SW79] K. R. Scherer and H. G. Wallbott, editors. *Die Funktionen des nonverbalen Verhaltens im Gespräch, in Nonverbale Kommunikation: Forschungsberichte zum Interaktionsverhalten*. Beltz Verlag, 1979.

[TA92] D. Traum and J. Allen. A speech acts approach to grounding in conversation. In *Proc. 2nd Int. Conf on Spoken Language Processing*, 1992.

[TBB+99] S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Haehnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Minerva: A second generation mobile tour-guide robot. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA'99)*, 1999.

[TBB+00] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Probabilistic algorithms and the interactive museum tour-guide robot Minerva. *Int. Journal of Robotics Research, Special Issue on Field and Service Robotics*, 19(11), 2000.

[TBC+02] C. Theobalt, J. Bos, T. Chapman, A. Espinosa-Romero, M. Fraser, G. Hayes, E. Klein, T. Oka, and R. Reeve. Talking to godot: Dialogue with a mobile robot. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS2002)*, 2002.

[TC06] E. Topp and H. I. Christensen. Topological modelling for human augmented mapping. In *Proc. Int. Conf. on Intelligent Robots and Systems*, 2006.

[TD98] D. Traum and P. Dillenbourg. Towards a normative model of grounding in collaboration. In *working notes, ESSLLI-98 workshop on Mutual Knowledge, Common Ground and Public Information*, 1998.

[Tha83] C. Thavenius. Referential pronouns in english conversation. In *Lund Studies in English*, number 64. Gleerup, 1983.

[THCSE06] E. A. Topp, H. Huettenrauch, H. I. Christensen, and K. Severinson-Eklundh. Bringing together human and robotic environment representations – a pilot study. In *Proc. Int. Conf. on Intelligent Robots and Systems*, 2006.

[Thó96] K. R. Thórisson. *Communicative Humanoids A Computational Model of Psychosocial Dialogue Skills*. Dissertation, Massachusetts Institute of Technology, 1996.

[Thó97] K. R. Thórisson. Gandalf: An embodied humanoid capable of real-time multimodal dialogue with people. In *Proc. 1st ACM Int. Conf. on Autonomous Agents*, 1997.

[Thó99] K. R. Thórisson. A mind model for multimodal communicative creatures and humanoids. *Int. Journal of Applied Artificial Intelligence*, 13(4–5), 1999.

[TLWF04] I. Toptsis, S. Li, B. Wrede, and G. Fink. A multi-modal dialog system for a mobile robot. In *Proc. International Conference on Spoken Language Processing*, 2004.

[TPJ$^+$02] N. Tomatis, R. Philippsen, B. Jensen, K. O. Arras, G. Terrien, R. Piguet, and R. Siegwart. Building a fully autonomous tour guide robot: Where academic research meets industry. In *Proc. Int. Symp. on Robotics*, 2002.

[TR00] M. Turk and G. Robertson. Perceptual user interfaces. *Communications of the ACM*, 43(3), 2000.

[TR02] D. Traum and J. Rickel. Embodied agents for multi-party dialogue in immersive virtual world. In *Proc. 1st Int. Conf on Autonomous Agents and Multi-agent Systems*, 2002.

[Tra94] D. Traum. *A Computational Theory of Grounding in Natural Language Conversation*. Dissertation, University of Rochester, 1994.

[Tra96] D. Traum. Conversational agency: The trains-93 dialogue manager. In *Proc. Twente Workshop on Language Technology: Dialogue Management in Natural Language Systems*, 1996.

[Tra97] D. Traum. Discourse resource initiative. standards for dialogue coding in natural language processing. In *Report No. 167, Dagstuhl Seminar*, 1997.

[Tra99] D. Traum. Computational models of grounding in collaborative systems. In *working notes of AAAI Fall Symp. on Psychological Models of Communication*, 1999.

[TRS04] D. Traum, S. Robinson, and J. Stephan. Evaluation of multi-party virtual reality dialogue interaction. In *Proc. 4th Int. Conf. on Language Resources and Evaluation*, 2004.

[TVS01] N. Tschichold, S. Vestli, and G. Schweitzer. The service robot MOPS: First operating experiences. *Robotics and Autonomous Systems*, 2001.

[VW96] M. T. Vo and C. Wood. Building an application framework for speech and pen input integration in multimodal learning interfaces. In *Proc. Int. Conf. on Acoustics Speech and Signal Processing*, 1996.

[WAB$^+$01] M. Walker, J. Aberdeen, J. Boland, E. Bratt, J. Garofolo, L. Hirschman, A. Le, S. Lee, S. Narayanan, K. Papineni, B. Pellom, J. Polifroni, A. Potamianos, P. Prabhu, A. Rudnicky, G. Sanders, S. Seneff, D. Stallard, and S. Whittaker. Darpa communicator dialog travel planning systems: The june 2000 data collection. In *Proc. 7th Euro. Conf. on Speech Communication and Technology*, 2001.

[Wah03] W. Wahlster. Smartkom: Symmetric multimodality in an adaptive and reusable dialogue shell. In *Proc. Human Computer Interaction Status Conference*, 2003.

[WDK$^+$05] S. Woods, K. Dautenhahn, C. Kaouri, R. Boekhorst, and K. L. Koay. Is this robot like me? links between human and robot personality traits. In *Proc. IEEE-RAS Int. Conf. on Humanoid Robots*, 2005.

[WFBS04] S. Wrede, J. Fritsch, C. Bauckhage, and G. Sagerer. An XML Based Framework for Cognitive Vision Architectures. In *Proc. Int. Conf. on Pattern Recognition*, 2004.

[WLKA97] M. Walker, D. J. Litman, C. A. Kamm, and A. Abella. Paradise: A framework for evaluating spoken dialogue agents. In *Proceedings of the 35th annual meeting of the association for computational linguistics*, 1997.

[WRP$^+$02] M. Walker, A. Rudnicky, R. Prasad, J. Aberdeen, E. Bratt, J. Garofolo, H. Hastie, A. Le, B. Pellom, A. Potamianos, R. Passonneau, S. Roukos, G. Sanders, S. Seneff, and D. Stallard. Darpa communicator: Cross-system results for the 2001 evaluation. In *Proc. Int. Conf. of Spoken Language Processing*, 2002.

[WSS94] J. H. Walker, L. Sproull, and R. Subramani. Using a human face in an interface. In *Proc. Conf. on Human Factors in Computing Systems*, 1994.

[Xml] Xml. http://www.w3.org/XML/.

[XR00] W. Xu and A. Rudnicky. Task-based dialogue management using an agenda. In *Proc. ANLP/NAACL 2000 Workshop on Conversational Systems*, 2000.