Towards Multimodal Communication with a Household Robot

P.Gieselmann, C.Fügen, H.Holzapfel, T.Schaaf*, and A.Waibel

Interactive Systems Lab, ILKD
Universität Karlsruhe
Am Fasanengarten 5
76131 Karlsruhe, Germany
{petra, fuegen, hartwig, tschaaf, waibel}@ira.uka.de

Abstract. This paper is about the multimodal dialogue between humans and robots in household environments. We explain the speech recognition and the understanding and dialogue management components. Above all, we describe how new words could be recognized and learned in the area of speech recognition and how emotion recognition could improve the dialogue processing. We made a first data collection with a very simple system which serves as a base for all our further developments. By means of this data collection, we enhanced our grammars and lexicons and we found efficient ways of improving the interaction between the different components in the system. We also discovered possibilities for a promising integration of different modalities such as gestures and emotion recognition.

1 Introduction

The biggest challenge for interaction with a household robot is its ease of use. Everybody should be able to use this robot without any initial training; it should work as a real housekeeper. Therefore, it is important that the user can interact with the robot in the same way as with other humans - via speech and gestures.

This kind of multimodal human-machine interaction facilitates the communication for the user of course, whereas it is quite challenging from the robot's point of view. For example, we have to cope with spontaneous speech, different dialects and even ungrammatical utterances which still have to be understood correctly by the system. Furthermore, the speech recognition has to deal with bad acoustic conditions in rooms with music in the background or with crying children for example and different speakers to whom the system should adapt very fast and easily.

The dialogue management component has to deal with elliptic utterances of the user which could only be resolved by adding information from other knowledge sources such as gestures for example. Therefore, we will explain how multimodal parsing is used.

* now with SONY International (Europe), Heinrich-Hertz Str. 1, 70327 Stuttgart, Germany

Besides, the robot has to cope with a nearly indefinite number of user utterances. Since it is impossible to construct a grammar or a concept hierarchy which really covers all the words the user will ever utter or all the tasks the user wants the robot to accomplish, it is important that the user can enhance the grammar of the robot by teaching it new concepts. But first of all we want to assure a basic coverage of the grammar and of the concept hierarchy. That's the reason why we made a data collection with several people to set up such a base system and get first user results.

The acceptance of such a robot in domestic environments strongly depends on the user's confidence in its reliability. It is important that the robot notices the emotional state of the human. If the user is for example already very angry, the robot should not tell him again and again that it did not understand him, but try to find other ways to get to know what the user wants it to do. Therefore, we also evaluate the possibility of including emotion recognition in the human robot dialogue.

This paper deals with speech recognition and understanding and dialogue processing. Multimodal communication in the form of gesture and emotion recognition is also explored. Section two gives an overview of our speech recognizer and how adaptation techniques are used to improve the recognition rate even in bad acoustic environments. The mechanisms for recognizing and learning new words are also explained. Section three deals with the dialogue management. Section four gives experimental details and results, and section five gives a conclusion and outlook.

2 Speech Recognition

In this section we describe the problems occuring when humans communicate with machines in a natural way. The problems are continuous spontaneous speech input, different speaker dialects, bad acoustic conditions, and infinitely growing vocabulary.

2.1 Janus and the Ibis Decoder

For speech recognition, we are using the Janus Recognition Toolkit (JRTk) [5], jointly developed at the University of Karlsruhe and at the Carnegie Mellon University, Pittsburgh. For decoding, we are using the Ibis single pass-decoder, which was recently developed at Karlsruhe [16]. Ibis uses less memory and allows higher recognition speed than the three pass search originally implemented in Janus. In addition, it provides the option of decoding along context free grammars (CFG) instead of statistical n-gram language models (LM).

The context-free grammar capabilities of the Ibis decoder allow input queries to be directly parsed during the decoding so that there is no need for other external parsers. Moreover, other modules such as a dialogue manager can control the decoding process by penalizing or excluding specific rules. It is also possible to load several different grammars in the decoder so that you can for example easily switch between different domain grammars without restarting the recognizer. All of these features help the decoder to deal with the problems generated by interaction with a household robot: Spontaneous speech and dialects, bad acoustic conditions, and unknown words.

2.2 Spontaneous Speech and Dialects

Compared to statistical n-gram language models, grammars have the disadvantage of an inadequate modeling of hesitations and other spontaneous effects. Speech or non-speech noises for example can occur anywhere in a sentence. To compensate, we model such noises as filler words in the decoder. During decoding the score is given implicitly to the decoder by a fixed value and not by the language model. In addition, we have chosen semantic instead of syntactic context-free grammars to model the system knowledge, because they are known to be more robust against ungrammaticalities in spontaneous speech and recognition errors [18].

The results of experiments in the domain of LingWear, i.e. spontaneous speech queries for a wearable linguistic assistant for tourists, are given in table 1. As the table shows, the sentence correct rate (SCR) and the recognition speed (RTF = real time factor) is about 20% better when using a grammar than with a statistical n-gram language model.

	LM	CFG
Word Accuracy	76.12%	77.29%
SCR	40.57%	51.64%
RTF on PIII, 1GHz	0.20	0.15
Memory Requirements	35 MB	35 MB
Vocabulary Size	2035	2035

Table 1. Comparison of a 3-gram LM and a CFG on ~ 250 Sentences

In spontaneous speech, most of the data is influenced by the dialect of the speaker. Therefore, we are working with pronunciation variants in the dictionary and with dialect dependent context decision trees in the recognizer. This allows us to automatically train and cluster mode-dependent acoustic models for different modes. You can use this method also with other modes than dialect, such as the speaking rate or the signal to noise ratio of a signal. This gives a 10% relative gain on German spontaneous speech data (GSST) [6].

2.3 Acoustic Conditions

A household robot has to cope with different kinds of environmental noise which are recorded by the robot's microphones. Moreover, the distance between the microphone and the user will produce a lower signal-to-noise ratio and more reverberations in the signal than close-talking recording conditions. All of these contribute to worse speech recognition accuracy.

To reduce these influences we are using a model-combination-based acoustic mapping (MAM) which was developed in our lab for speech recognition in car environments [17]. This method combines noise compensation together with adaptation techniques. Figure 1 shows the results of read speech at different microphone distances. Combining maximum likelihood linear regression (MLLR) together with MAM gives a significant gain for distant read speech.

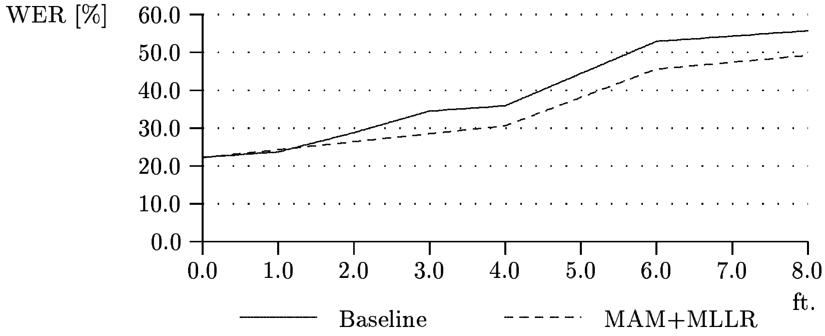

Fig. 1. Results of 9 speakers on Distant Read Speech Data

2.4 Detection and Extraction of Unknown Words

Another challenge for speech recognition in our experiments is the recognition of unknown words. Since the recognizer vocabulary cannot cover all of the words a user may say to the robot, the robot should at least be able to learn unknown words (OOV = out of vocabulary) together with their pronunciations and meanings so that it will be possible for the user to teach the robot new words.

Therefore, it is at first necessary to detect an unknown word in a speech signal. After this, the pronunciation of the word must be extracted, together with its semantics so that the new word can be added to the vocabulary and the language model.

Detection of Unknown Words We are using so called head-tail structures of generalized word models to detect unknown words. Generalized word models consist of a specifically modeled head and a general tail (see figure 2). The head consists of a sequence of regular phone models and the tail consists of a sequence of generalized phone models [15]. A generalized phone is an acoustic unit that models all or at least a subset of the phone inventory of a language. A generalized word is a word that contains one or more generalized phones. The idea behind these head-tail structures is that the likelihood of such a structure in an OOV situation is higher than the likelihood of any well-trained word from the recognizer vocabulary.

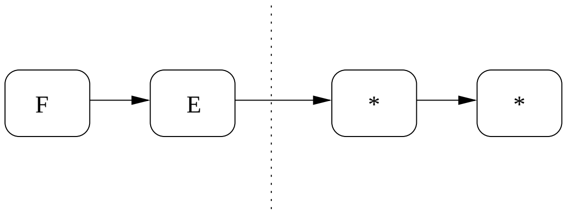

Fig. 2. Head-Tail Structure

Table 2 shows significant gains for read and spontaneous speech. WCE is the word class error rate, which is equal to the word error rate after replacing unknown words in the reference with <UNK>. Generalized words in the hypothesis are also replaced with <UNK>. Recall (REC) is the percentage of how many of the available unknown words in the data are detected and precision (PRC) is the percentage of how many of the recognized unknown words are correctly detected.

	WCE	REC	PRC	WCE	REC	PRC
BASE	38.9%			22.6%		
GW780	21.1%	59%	100%	22.2%	57%	77%
CHEAT	0.4%	97%	100%	21.9%	74%	100%

Table 2. Results on Read and Spontaneous German Speech Data

Extraction of the Pronunciation and the Semantics To allow the recognizer to work with a new word, the pronunciation of the word has to be generated. If the spelling of the word is known, this can be done by looking into a large background dictionary or by using letter-to-sound rules. But for natural communication the robot should be able to generate the pronunciation automatically, without asking the user for help. Therefore we are using a phone recognizer, which runs on the detected region of the unknown word in the speech signal. The phone recognizer achieves a phone error rate of 35%.

The advantage of using head-tail structures is the possibility to add them as words to language model classes or grammar rules. This implies that one head-tail structure could belong to several different classes or rules. If such a head-tail structure is recognized, the semantic information is directly given by the class membership of the head-tail structure.

3 Dialogue Management

Once the speech is recognized, the robot must understand what the user has said so that it can react in a reasonable way. Therefore, the dialogue component analyses the utterance of the user and creates a semantic representation of it. This semantic representation contains a task the robot can accomplish and all the parameters which are necessary for this task. An example is "take something," and the parameters are the object which should be taken and the place from where it should be taken. If this semantic representation which is extracted from the user's utterance already specifies the task completely, instructions are passed to the robot itself, for example moving to the place where the thing is which should be taken. But if some information, like the object which should be taken, is still missing, the dialogue component will ask the user for it. In this way, it is assured that only complete task descriptions with all the necessary parameters are sent to the other processing modules of the robot.

It is important that the user can express his wishes in different ways. It does not matter whether he says "I would like to get the cup from the board" or "pick up the cup from the board". In both cases, the robot recognizes the dialogue goal "to get something" and the parameters "cup" and "board".

3.1 Architecture of the Dialogue Manager ARIADNE

In this project, we use the language and domain independent dialogue manager ARIADNE developed by Matthias Denecke at Carnegie Mellon University in Pittsburgh [1]. One of the reasons why we use this dialogue system is that the user can formulate his commands in different ways so that he is not restricted to one way of saying something. It achieves this by using typed feature structures rather than a frame-based approach which puts much more restrictions on the user utterances.

In addition, ARIADNE is specifically designed for rapid prototyping. Only the domain and language dependent components have to be implemented for a new application, since the general concepts are already available. This is made possible by vectorized context-free grammars and inheritance mechanisms. General input and output mechanisms and methods for evaluating the dialogue state are already implemented and can be used in the actual application.

Furthermore, multidimensional feature structures are used [3]. This means that not only semantic information can be saved at the nodes of the tree, but also information on the input modality and for example also confidence measures of this input. In this way, it is possible to ask the user specifically for the words which could only be recognized with a very low confidence measure for example.

The dialogue manager uses different kinds of task and domain dependent resources (see figure 3): an ontology, a specification of the dialogue goals, database rules, a grammar and generation templates. There is also a dialogue strategy which decides how new information could be interpreted and integrated.

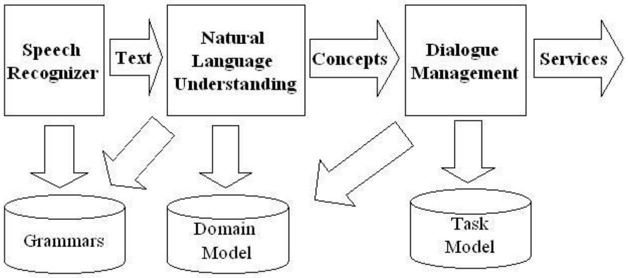

Fig. 3. The Dialogue Manager and its Resources

Dialogue Grammar and Domain Model First of all, the input of the user is parsed by means of a semantic grammar which also consists of some syntactic information such as which word is a noun phrase, which is a verb, etc. Although the semantic information is domain- and application-dependent, the syntactic information is independent. According to the principles of rapid prototyping, domain independent information can be reused; other dialogue managers do not allow for this reuse because of the mixture of semantics and syntax. In ARIADNE, the separation of syntactic and semantic information is implemented by means of the vectorized context-free grammars which consist of non-terminals of n-dimensional vectors of partially organized elements [4].

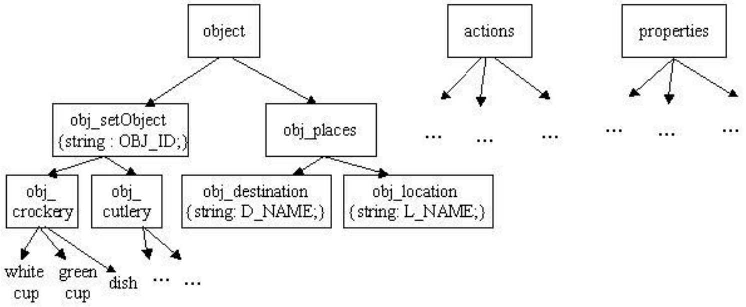

Fig. 4. Part of the ontology. The generic concepts are linked to the application specific information here.

Because of this separation, syntactic information can be reused. Therefore, the construction of complex noun phrases such as "the red cup" can be found in a general part of the grammar, while the semantic instantiations of the objects are in the domain-dependent part (see figure 4).

Furthermore, it is in this way possible to combine the domain model with the actual grammar. The domain model determines which concepts the system knows and how they can be combined. It is built up as an ontology with objects, actions and properties which could inherit from each other (see figure 4). Therefore, it is also possible to access the domain independent general ontology which consists of concepts such as different speech acts and general goals, objects and properties from which specific objects, actions and properties could then inherit in the domain dependent part.

The grammar used by the dialogue manager can be converted into a non-vectorized context-free grammar and used in this way by the speech recognizer so that both components use the same linguistic knowledge base.

Task Model The task model specifies dialogue goals which correspond to the services the robot can execute. A dialogue goal can be seen as the description of a form which is filled by means of the dialogue between human and machine [1].

This means that the dialogue goals are specified by the information the user gives in the discourse and that they consist of objects, actions and properties which are defined in the ontology. Therefore, the dialogue goals are the connection between the domain model and the services the dialogue manager can execute.

If a dialogue goal is recognized, the dialogue manager searches for the corresponding parameters in the discourse, such as objects, properties and actions. If the feature structure is still underspecified, a clarification dialogue is initiated.

Generation Templates All the information from the robot to the user is given in natural language through the use of generation templates. In these generation templates, the dialogue state determines what the dialogue manager tells the user in which situation. The dialogue state is defined by the information in the dialogue goals as shown in figure 5 [1].

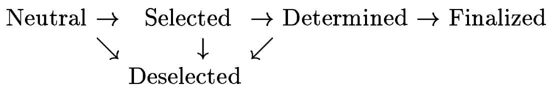

Fig. 5. Dialogue States and their Transitions

At the very beginning of a dialogue, the dialogue state is neutral; then some dialogue goals are selected, and the state becomes selected. A selected dialogue goal becomes determined if it is the only one which is selected. The dialogue goal becomes finalized when all information which is specified in the dialogue goal is available in discourse - this means that all the variables are specified.

These generation templates determine on one hand what the robot asks the user in which situation and on the other hand, what the dialogue manager expects as an answer and how this answer is integrated.

The robot also uses context knowledge so that it does not ask the user for information which could be deduced. For example, if only one cup is in the room, the robot does not ask the user which cup to take, but simply takes the cup.

Databases The database contains objects and their properties as they can be found in the environment of the robot. In this way, the dialogue manager can search for different instances of cups and their places in such a database and if there are for example different cups in the room, a clarification question is generated to ask the user which cup the robot should take.

Dialogue Strategy The dialogue strategy defines how different kinds of information are evaluated in a specific dialogue state. This means that here the general proceeding mechanisms are specified. The dialogue strategy consists of different interaction patterns which define how information can be added and deleted in the discourse.

This strategy can normally be reused for different applications. It is only necessary to adapt it if we want to integrate additional modalities, for example emotion recognition.

3.2 Multimodal Parsing

Since human human communication normally consists of speech and gestures, we also want to include gestures in our interaction. The two modalities, speech and gestures, can be integrated by multimodal parsing. We use time stamps to unify them. This means that when speech and gestures occur more or less at the same time, they should be evaluated together.

In this case, it is also possible to use the confidence measures from both speech and gesture recognizers to determine if clarification questions are necessary. For example, if the confidence measures for the speech recognition and the ones for the gesture recognition are both very low, but both refer to the same object, the overall confidence measure is increased. Otherwise, the robot can ask the user to clarify which object was intended.

An additional benefit of multimodality is that ambiguities can be resolved by sensor fusion. When the user for example says "Get this cup" and points at the same time to a cup, then the system knows which cup is meant although different cups are in the room. The unification of the semantics of speech and gestures as explained in [10] promises good results.

3.3 Emotions as a Parameter of the Dialogue Strategy

Emotional cues are used to obtain some kind of user model for a better adaptation to the user's wishes. So far, we have focused on the user's emotions and did not take into account the possibility of letting the robot express emotions. Our model of the user's emotional state, allows us to choose different dialogue strategies depending on the state of the user. Therefore we use the Affective Dialogue framework which is fully described in [9].

To use the dialogue system efficiently in different environments, robust emotion recognition is required. The features used by the emotion recognizer must be obtained with the available sensors. Besides, we only need emotions carrying application-relevant information.

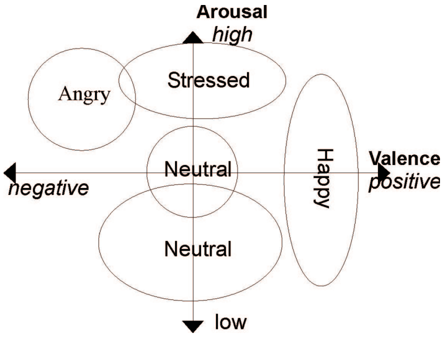

Fig. 6. Arousal - Valence - Level

The Nature of Emotions In literature different work can be found that gives definitions of emotions and describes their characterizations. Also different work has been done to recognize emotions (e.g. [13]). It is commonly agreed that emotions have physical and cognitive aspects. Accordingly, there are two main approaches to recognize emotions. To classify emotions according to the cognitive model, an extensive world model is needed that models wishes, goals and possibly fears of the user (see [12]). However, situations of every day life are generally too complex to model. A different approach measures changes of the human's physical state; this includes speech, heart rate, mimics, skin conductivity, etc. [13]. We prefer this approach, since it seems to be more domain-independent. Measuring the physical state is most commonly combined with the arousal-valence model to characterize emotions in a two-dimensional plane [11], see figure 6. Figure 7 shows a possible discretization as required by the dialogue system. However, the dialogue system abstracts from the actual emotion model [8]. Thus, an extended emotion recognizer may be used in the future which will be able to process further cognitive cues, if they turn out valuable and improve the model's accuracy.

The Affective Dialogue System's Architecture Figure 8 shows the included components of the dialogue system in a dataflow diagram. The input

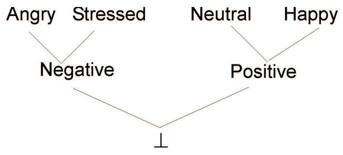

Fig. 7. Discretization

is completely speech based. The hypothesis of the speech recognizer and the emotion recognizer are converted into a semantic representation and sent to the dialogue system. The semantic representation of the spoken utterance is annotated with emotional information using the Multidimensional Typed Feature Structures formalism [8], mentioned earlier.

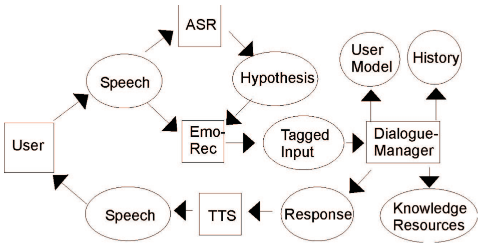

Fig. 8. Data Flow

Extending the Dialogue Strategies In order to integrate emotion recognition in the dialogue strategy, we added new emotion variables to the dialogue state. The following three variables are currently used to represent emotion values:

- UserEmotion: represents the current emotional state of the user.
- UserEmotionTrend: represents the trend the user's emotion seems to take.
- SystemEmotion: represents the system's strategy for reacting to the user's emotion values.

System Strategies Before starting the data collection we had some intuitive ideas how the robot should behave. The ideal behavior of the robot varies, depending on the person interacting with it. The robot should not categorically imitate human behavior; for example, it is not clear that the robot should show anger. In fact, angry reactions of the human interactor might indicate wrong behavior of the robot. In response, it is the dialogue system's task to validate the previously executed action by re-asking the user.

4 Data Collection

The first dialogue system we implemented focused on a simple task: Setting the table. This system serves as our baseline and works with push-to-talk interaction. This system can be easily enhanced, and later versions will have more complex interfaces and scenarios.

The motivation for this data collection was twofold: On one hand, we wanted to get data for enhancing the grammar and the lexicon and to watch why and when users show emotions and how the system should react in these cases. We also wanted to generally test the system and its different components and see how the users can manage the whole task and how they cope with situations where the system did not understand them. The results of the data collection are then used in an interactive development cycle to see where the system could be easily improved for better interaction with the user. The three-stage development cycle consists of pretests to ensure that the whole architecture works and that the test subjects could generally manage the task, a test with a simple task and another test with a more complex task for the user.

Since the real robot was not yet available for the data collection, we implemented a simulation environment in which the user can see the robot in the kitchen on-screen, along with the table to set and a board with the available dishes and cutlery on it (see figure 9).

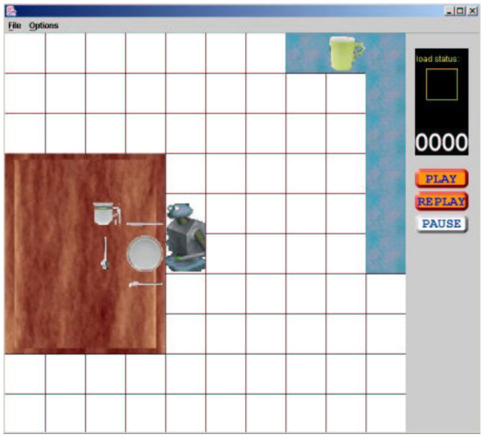

Fig. 9. The Simulation Program

The users were asked to make the robot set the table for one person. They were not given any detailed instructions about what to say to the robot. This allows them to explore everything on their own, which corresponds well to a real kitchen scenario.

Most of the test persons were filmed on video so that we have additional data for the emotion recognition. A small interview was done after the test to get their general impression of the system; they were asked what they liked and what they didn't like and what they think should be improved.

We implemented a small dialogue grammar with different dialogue goals so that the robot can go somewhere, take something from somewhere, put something in a given place, say hello to the user, ask the user for instructions on how to set the table and say thanks to the user. The robot was not able to take more than one object at a time and it informed the user of this when necessary. The robot's world knowledge consisted of a database with all the objects on the table, on the board, and in the robot's possession. In this way, the robot knows where which object can be found.

4.1 Wizard of Oz Experiment

The data collection was done by means of a Wizard of Oz experiment so that the user is not limited to the grammar and the vocabulary which is already implemented. We tried to interfere as little as possible during the experiment in order to see whether the user could still accomplish the task by himself.

For this experiment, we set up a client server architecture. On the client side, the speech recognizer, the simulation program and the speech generation can be found. On the server side, there is the dialogue manager and the java interface for the experimenter as shown on the figure below (see figure 10).

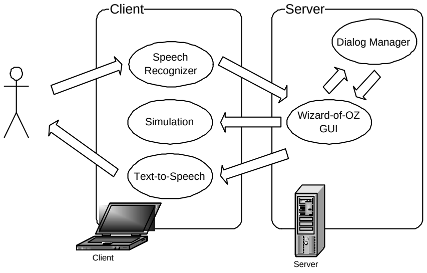

Fig. 10. Client-Server-Architecture

In pretests, we noticed that the speech recognition was quite bad at the beginning because the test persons uttered only very small sentences, preventing smooth adaptation by the recognizer to the new user. Therefore, we added a small unsupervised adaptation phase for the speech recognizer with four sentences to be read by all the test persons at the beginning of every test.

On average, the six pretest subjects needed 26 turns to make the robot set the table, which is quite fast given the fact that a minimum of 10 turns would be necessary to get the five different objects from the board and then put them on the table one by one.

After the pretest we also added some new terms to the vocabulary used by these test persons and then started the real test.

As shown in the table below (see table 3), the adaptation phase improves the recognition significantly: Without any adaptation, we had a turn error rate of 83%, but with this short adaptation phase, the turn error rate decreased to approximately 53%.

	without Adaptation	with Adaptation
Avg. # of Turns	26	22
Avg. # of Wrong Recognized Turns	22	12
Turn Error Rate (in %)	83	53

Table 3. Turn Error Rates with and without Unsupervised Adaptation of the Speech Recognizer

4.2 Results

Our data collection showed that most of the people were quite happy with the system, as table 8 shows. When the users did have problems, most of them complained about the bad speech recognition. All users managed to make the robot set the table. On average the test persons need 22 turns for this task and we observed that the faster they were, the more they liked the system.

	Minimum	Average	Maximum
# of Turns	13	22	32
# of Wrong Recognized Turns	5	12	19
Turn Error Rate (in %)	38	53	59

Table 4. Turn Error Rates

Table 4 shows that there is a wide range in the number of turns from 13 to 32. This can be explained by the fact that the turn number depends very much on the result of the speech recognition. One reason for these recognition problems is that not all the vocabulary the test persons used is already in the lexicon of the speech recognizer, leading to bad results. A solution is to use n-gram language models combined with grammars, which we will evaluate soon.

Speech recognizer robustness is of special importance because we see lots of variation in expressing the same meaning: All together, there were more than ten different ways to tell the robot to get something.

Another solution for these speech recognition problems is that more interaction between the speech recognizer and the dialogue manager is necessary in the future. For example, if the user would probably answer by "yes" or "no" to a clarification question from the dialogue manager, other hypotheses should be penalized by the speech recognizer.

The overall word error rate of the speech recognizer was 79.4%. Table 5 shows the wide variation in accuracy across speakers, ranging from 42.2 to 85.5%. This can be explained by the wide range of vocabulary used by different test persons.

	Minimum	Average	Maximum
Word Error Rate	42.2	79.4	85.5

Table 5. Speaker-Specific Word Error Rates

Uncovered Expressions The test subjects used lots of expressions not covered by our system so far. Some of them refered to the way the table should be set, trying to make the robot move everything to another part of the table. Some tried to make the robot put a drink in the cup, which was also not covered in this scenario. In this way, the users tested what the robot can do for them.

Furthermore, uncovered expressions occurred when the robot did not understand what the user said. In reaction, users asked the robot whether it was still awake, why it didn't understand them, etc. Most of these expressions are now included in the system so that the user gets at least some feedback; nevertheless, there will still be users who invent utterances not covered by the system so far. Table 5 shows the improvement in the system's utterance coverage in the different stages of the development cycle.

	Minimum	Average	Maximum
Pretest (in %)	18.75	44.9	79.17
Simple Task (in %)	23.08	35.6	53.33
Complex Task (in %)	13.51	31.85	69.15

Table 6. Uncovered Expressions

In the future, we want to solve this problem of out-of-domain sentences where the commands uttered by the user were not part of the table-setting scenario by integrating learning techniques in the dialogue manager so that it is not only able to learn new words, but also new concepts.

Human Robot Interaction The interaction between the human and the robot went quite well. When a user tries to make the robot do something it cannot, it tells the user. For example, when the user asks the robot to put the knife on the table and the robot does not have anything in its hands, it will inform the user that he has to tell the robot to get something before it can be placed on the table. In this way, the users learn very fast what is possible. Unfortunately, this means that the users adapt to the system, although our intention was that the robot should adapt to the human and not vice versa. Nevertheless, this adaptation effect seems to be above all due to the fact that our system was still quite small and the task quite simple. We think that in the more complex system which we are now building, this effect will be of minor importance; we will evaluate this in further user studies.

In this scenario, we focused on step-by-step instructions, meaning that the user has to tell the robot every step in setting the table. All this information is then gathered by the robot and a context model is build up so that the robot can later set the table by itself because it has learned all the necessary steps. In this way, the robot will be able to execute not only simple instructions, but also complex commands which are created out of these step by step instructions after having learned the context model.

Since some of the users still complained that the robot does not say which words it did not understand, we want to evaluate the confidence measures from the speech recognizer in a more sophisticated way so that the robot can ask the user again in all the cases where the confidence is too low.

Integrating Discourse Information At the moment, we cannot resolve pronouns with the dialogue system we use. Most of the users noticed that during the data collection and then tried to avoid pronouns, but they complained about it in the interview. In the future, we will therefore integrate anaphora resolution by means of the information which is already available in discourse.

In the future, context information also has to be taken into account to a higher degree during dialogue processing. This will allow expressions such as ”back” to be resolved. To accomplish this, we will implement a context model in the dialogue manager which could resolve these expressions.

Simulation Specific Problems Some problems we encountered were specific to the simulation because the user does not see for example how difficult it is for the robot to take more than one thing at a time. Nearly all the users tried to make the robot take two objects at a time, which was not possible in our simulation. Many users complained in the interview about that.

Using more Complex Tasks After this first test, we decided to make the task more difficult for the user by adding another cup which was also on the board, but which should not be put on the table. This task was especially challenging for the test persons because the robot's default behaviour was to pick up the wrong cup. In this way, we wanted to provoke more emotional reactions of the test persons and see how they cope with more difficult situations.

As expected, this task leads to an increase of the average turns to 54, because the users got this cup and then had to put it back again. As the table below (see table 7) shows, the users needed more turns to accomplish the more complex tasks, but at the same time the turn error rate decreased. This is above all due to the fact that we added all the missing vocabulary from previous tests which lead to a better recognition rate.

	Simple Task	Complex Task
Avg. # of Turns	22	54
Avg. # of Wrong Recognized Turns	12	21
Turn Error Rate (in %)	53	38

Table 7. Turn Error Rates for the Simple and the Complex Task

User Satisfaction As shown in the table below, the users were quite happy with the system. Even in the pretest more than half of the users were satisfied (happy or neutral) with the system. During the whole user test, the system has been gradually adapted to the user needs so that the user satisfaction increased up to 50% happy users.

	unhappy	neutral	happy
Pretest (in %)	44	28	28
Simple Task (in %)	25	25	50
Complex Task (in %)	17	33	50

Table 8. User Satisfaction

Results of Using Emotions in Dialogue Processing One focus of the data collection was to find out when and how users show emotions, along with how the system should react in these situations. In the pretest the two main emotions of the users were frustration/anger and happiness/fun (see table 8). In the interview after the test, most of the users who responded angrily to the system also reported that they were frustrated because the system did not understand them. This was either because the speech recognizer did not produce the correct hypotheses or because the dialogue manager could not recover from an unwanted state. Happiness was mostly because the users found the system to be “cute”.

We also discovered that the push-to-talk interaction limits the range of emotions that could be recognized by the system. Different (visual) expressions like shrugging or frowning could be observed when unexpected things happened or the robot misinterpreted commands; some users also used their voice to express astonishment, joy or disappointment. However, these events occurred during the speech breaks. All users seemed to rethink their next action before pushing the button to talk to the robot. This produced a small time slot between the users' first (intuitive) reactions and the spoken command to the robot. The utterances spoken to the robot seemed to be spoken without changing attitude. This means that when the user started one utterance with an angry voice, he also completed the utterance with an angry voice.

Seven randomly-selected users were asked to take part in a competition, with the goal to complete the task as quickly as possible. During their task a timer was running and counting seconds. The numbers of the counter were well visible as can be seen in figure 9. These users were more task-focused than the non-timed users, which means that they didn't try many different ways to interact with the system. However, we could not say that these users were more stressed than others or that they reacted more angrily to errors of the dialogue system. Three of these users reached their goal very fast because their input was corrected by the Wizard-of-Oz control and the others interacted with the system without interference. As we had expected, the fast ones were happy that they had completed the task successfully. The slower users were frustrated about the bad result, but we could not say that they were more angry or more frustrated than non-timed users who got the same bad results.

It was interesting to see that people who complained about bad speech recognition also disliked other parts of the system, such as the simulation environment that in turn was appreciated by most of the users. This suggests that feelings about one bad result also influence feelings about other parts of the system that are independent of the bad result. Thus, it is very important to prevent the user from getting angry.

We are currently developing an emotion recognizer that was not ready yet for the evaluation. Hence, our analysis of user emotions is based solely on experimenter observations. Some strategies will be tested in the near future using the Wizard-Of-Oz system in place of the emotion recognizer.

5 Conclusion and Outlook

In this paper, we gave an overview of our current research in multimodal human robot interaction. Speech recognition and understanding and dialogue management have been explained in detail and we mentioned how the components work together. In addition, mechanisms for recognizing and learning new words have been developed. In the future, we would like to extend our research to learning of grammar rules, ontology concepts and whole dialogue goals so that the newly-learned words could be integrated in the whole environment easily.

Aspects of multimodal parsing have only been shortly mentioned, but they will be an important part of our future research.

The importance of integrating emotion recognition into human robot interaction has been explained and we will further investigate how this can be done in some more detail.

The results of our data collection showed that even with this simple system the users could manage their tasks well and were content with the whole system. We plan to enhance this system by improving the interaction between the dialogue manager and the speech recognizer, using confidence measures in the dialogue manager more extensively and using more effectively the single grammar shared by both components. We will also include additional modalities, such as gestures, to resolve ambiguities and emotion recognition for a more efficient human robot interaction. Therefore, we would like to make further data collections with more complex tasks and with gesture input, to get to know whether this additional input facilitates the task for the user and in which way both modalities could be successfully integrated.

6 Acknowledgement

This work is part of the Sonderforschungsbereich (SFB) No. 588 *"Humanoide Roboter - Lernende und kooperierende multimodale Roboter"* at the University of Karlsruhe. The SFB is supported by the Deutsche Forschungsgemeinschaft (DFG = "German Research Foundation").

References

1. Denecke, M.: Generische Interaktionsmuster für aufgabenorientierte Dialogsysteme. PhD Thesis. University of Karlsruhe, 2002.
2. Denecke, M.: Rapid Prototyping for Spoken Dialogue Systems. In: Proceedings of the 19th International Conference on Computational Linguistics, Taiwan, 2002.
3. Denecke, M., Yang, J.: Partial Information in Multimodal Dialogue. In: Proceedings of the International Conference on Multimodal Interfaces, 2000.
4. Denecke, M.: Object-oriented Techniques in Grammar and Ontology Specification. In: Proceedings of the Workshop on Multilingual Speech Communication, 2000.
5. Finke, M., Geutner, P., Hild, H., Kemp, T., Ries, K., Westphal, M.: The Karlsruhe-Verbmobil Speech Recognition Engine. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing, ICASSP-97, Munich, Germany, 1997.
6. Fügen, C., Rogina, I.: Integrating Dynamic Speech Modalities into Context Decision Trees. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing, ICASSP-00, Istanbul, Turkey, June 2000.
7. Fügen, C., Westphal, M., Schneider, M., Schultz, T., Waibel, A.: LingWear: A Mobile Tourist Information System. In: Proceedings of the Human Language Technology Conference, HLT-2001, San Diego, March 2001.
8. Holzapfel, H., Fügen, C., Denecke, M., Waibel, A.: Integrating Emotional Cues into a Framework for Dialogue Management. In: Proceedings of the 4th International Conference on Multimodal Interfaces, 2002.

9. Holzapfel, H.: Emotionen als Parameter der Dialogverarbeitung. Diploma Thesis, University of Karlsruhe, 2003.
10. Landragin, F.: The Role of Gesture in Multimodale Referring Actions. In: Proceedings of the 4th International Conference on Multimodal Interfaces, 2002.
11. Lang, P.J., Bradley, M.M., Cuthbert, B.N.: Emotion, Attention and the Startle Reflex. Psychological Review, Vol. 97, No. 3, 1990.
12. Orthony, A., Clore, G.L., Collins, A.: The Cognitive Structure of Emotions. Cambridge University Press, Cambridge, MA, 1988.
13. Picard, R.: Affective Computing. The MIT Press, 1997.
14. Rogina, I., Schaaf, T.: Lecture and Presentation Tracking in an Intelligent Meeting Room. In: Proceedings of the International Conference on Multimodal Interfaces, ICMI-0 2, Pittsburgh, USA, October 2002.
15. Schaaf, T.: Detection of OOV Words using Generalized Word Models and a Semantic Class Language Model. In: Proceedings of the EUROSPEECH-01, Aalborg, Denmark, September 2001.
16. Soltau, H., Metze, F., Fügen, C., Waibel, A.: A One pass- Decoder based on Polymorphic Linguistic Context Assignment. In: Proceedings of the Automatic Speech Recognition and Understanding Workshop, ASRU-2001, Madonna di Campiglio, Trento, Italy, December 2001.
17. Westphal, M., Waibel, A.: Model-Combination-Based Acoustic Mapping. In: Proceedings of the International Conference of Acoustics, Speech and Signal Processing, ICASSP-01, Salt Lake City, USA, May 2001.
18. Woszczyna, M., Broadhead, M., Gates, D., Gavalda, M., Lavie, A., Levin, L., Waibel, A.: A Modular Approach to Spoken Language Translation for Large Domains. In: Proceedings of AMTA-1998.