



Evaluating AdApt, a multi-modal conversational, dialogue system using PARADISE

Anna Hjalmarsson

Examiner: Rolf Carlson


Approved..... Examiner:
(signature)



Stockholm
November
2002


Master's Thesis in Speech Technology

Department of Speech Music and Hearing
KTH Royal Institute of Technology
100 44 Stockholm

| | |
|--|--|
|  <p data-bbox="186 451 430 630"> KTH Department of Speech, music and hearing </p> | <p data-bbox="462 231 958 262">Master's Thesis in Speech Technology</p> <p data-bbox="462 315 1136 493"> Evaluating AdApt, a multi-modal conversational, dialogue system, using PARADISE </p> <p data-bbox="462 577 706 609">Anna Hjalmarsson</p> |
| <p data-bbox="186 703 341 766"> Approved 2002-12-12 </p> | <p data-bbox="462 703 592 745"> Examiner Rolf Carlson </p> <p data-bbox="941 703 1071 745"> Supervisor Jens Edlund </p> |

Abstract

This master's thesis presents experiences from an evaluation of AdApt, a multi-modal, conversational dialogue system using PARADISE, PARAdigm for Dialogue System Evaluation, a general framework for evaluation. The purpose of this master's thesis was to assess PARADISE as an evaluation tool for such a system. An experimental study with 26 subjects was performed. The subjects were asked to interact with one of three different system versions of AdApt. Data was collected through questionnaires, hand tagging of the dialogues and automatic logging of the interaction. Analysis of the results suggests that further research is needed to develop a general framework for evaluation which is easy to apply and can be used for varying kinds of spoken dialogue systems. The data collected in this study can be used as starting point for further research.

| | |
|---|---|
|  <p>Institutionen för tal musik och hörsel</p> | <p>Examensarbete i talteknologi</p> <p>Att utvärdera AdApt, ett multimodalt konverserande dialogsystem, med PARADISE</p> <p>Anna Hjalmarsson</p> |
| <p>Godkänd 2002-12-12</p> | <p>Examinator Rolf Carlson</p> <p>Handledare Jens Edlund</p> |

Sammanfattning

Syftet med det här examensarbetet är att evaluera det multimodala, konverserande, dialogsystemet AdApt med PARADISE, PARAdigm for Dialogue System Evaluation, ett generellt ramverk för evaluering. Målet med utvärderingen var att undersöka hur väl lämpad PARADISE är som metod för att utvärdera denna sorts dialogsystem. För att undersöka detta utfördes en experimentell studie med 26 försöksdeltagare. Försöksdeltagarna fick uppgiften att interagera med en av tre olika versioner av AdApt. Data samlades in i form av frågeformulär och automatisk loggning av interaktionen med systemet. Dialogerna märktes sedan även upp för hand. En analys av resultaten föreslår att det behövs ytterligare forskning kring bland annat definition av uppgifter, val av mått och multimodala dialogsystem för att skapa ett lätt använt, användbart och generellt ramverk för utvärdering av dialogsystem

Contents

| | |
|--|-----------|
| 1 INTRODUCTION | 9 |
| 1.2 PURPOSE AND METHOD | 9 |
| 2 THEORETICAL BACKGROUND | 11 |
| 2.1 SPOKEN DIALOGUE SYSTEMS | 11 |
| 2.1.1 COMPONENTS OF SPOKEN DIALOGUE SYSTEMS | 11 |
| Speech Recognition..... | 11 |
| Language understanding | 12 |
| Dialogue Management | 12 |
| External communication | 13 |
| Response Generation..... | 13 |
| Speech output | 13 |
| 2.1.2 DIALOGUE MANAGEMENT STRATEGIES | 13 |
| Finite state based systems | 14 |
| Frame-based systems..... | 14 |
| Agent-based systems..... | 14 |
| 2.2 ERRORS IN HUMAN-MACHINE DIALOGUE | 16 |
| 2.2.1 SETTING EXPECTATIONS | 16 |
| 2.2.2 MISUNDERSTANDINGS AND NON-UNDERSTANDING | 16 |
| Misunderstandings | 17 |
| Non-understanding | 18 |
| 2.3 MULTI-MODAL SPOKEN DIALOGUE SYSTEMS | 20 |
| 2.3.1 MULTI-MODALITY | 20 |
| 2.3.2 SPEECH AND GESTURE | 20 |
| Multi-modal input | 21 |
| 2.3.3 ANIMATED SYNTHETIC FACES | 21 |

| | |
|---|-----------|
| 2.4 EVALUATION | 23 |
| <hr/> | |
| 2.4.1 WIZARD OF OZ | 23 |
| 2.4.2 SYSTEM IN THE LOOP | 23 |
| 2.4.3 OBJECTIVE METRICS | 24 |
| 2.4.4 SUBJECTIVE METRICS | 24 |
| 2.4.5 A GENERAL FRAMEWORK FOR EVALUATION | 25 |
| 2.4.6 EVALUATION OF MULTI-MODAL SYSTEMS | 26 |
| | |
| 2.5 PARADISE | 27 |
| <hr/> | |
| 2.5.1 TASK DEFINITION IN PARADISE | 27 |
| 2.5.2 MEASURING TASK SUCCESS | 28 |
| The Kappa coefficient | 29 |
| 2.5.3 DIALOGUE COSTS | 30 |
| 2.5.4 THE PERFORMANCE FUNCTION | 30 |
| | |
| 2.6 ADAPT A MULTI-MODAL CONVERSATIONAL DIALOGUE SYSTEM | 32 |
| <hr/> | |
| 2.6.1 ARCHITECTURE | 32 |
| 2.6.2 RESEARCH GOALS | 32 |
| 2.6.3 DIALOGUE STRATEGIES IN ADAPT | 33 |
| 2.6.4 TURN-TAKING GESTURES AND HOURGLASSES IN A MULTI-MODAL DIALOGUE SYSTEM | 34 |
| | |
| 2.7 USING PARADISE ON ADAPT | 35 |
| <hr/> | |
| 2.7.1 USING OPEN TASKS | 35 |
| Multi-modality | 36 |
| Choice of metrics | 36 |
| 2.7.2 PURPOSE | 36 |
| | |
| 3 METHOD | 37 |
| <hr/> | |
| 3.1 PRE-TESTS | 37 |
| <hr/> | |
| 3.2 EXPERIMENTAL DESIGN | 37 |
| <hr/> | |

| | |
|--|-----------|
| 3.3 EQUIPMENT | 38 |
| 3.3.1 ADAPT | 38 |
| 3.3.2 DAT-RECORDER | 38 |
| 3.3.3 VIDEO CAMERA | 38 |
| 3.4 DATA COLLECTION | 39 |
| 3.4.1 THE METRICS | 39 |
| Dialogue Costs | 40 |
| User Satisfaction | 41 |
| 3.4.2 HAND LABELLING | 43 |
| Task definition..... | 43 |
| Task Success | 45 |
| 4 RESULTS | 49 |
| 5. DISCUSSION | 53 |
| 5.1 THE PERFORMANCE OF PARADISE | 53 |
| 5.1.2 DEFINITION OF TASKS AND TASK SUCCESS | 54 |
| 5.1.3 MULTI-MODALITY | 57 |
| 6 FUTURE RESEARCH | 59 |
| 7. ACKNOWLEDGEMENTS | 61 |
| 8 REFERENCES | 63 |
| BOOKS | 65 |
| 9 APPENDIX | 67 |
| 9.1 CONFUSION MATRIX | 67 |
| 9.2 EXPERIMENTAL SET-UP | 68 |

| | |
|---|-----------|
| 9.3 DIALOGUE COST METRICS | 69 |
| 9.4 USER SURVEY | 70 |
| 9.5 SYSTEM INTRODUCTION | 71 |
| 9.5 MULTIVARIATE LINEAR REGRESSIONS FOR THE THREE SYSTEM CONTRIBUTIONS | 72 |
| 9.5.1 GESTURES | 72 |
| 9.5.2 HOUR-GLASS | 73 |
| 9.5.3 NO GESTURES | 74 |

1 Introduction

To create a conversational computer has been a challenging goal in both artificial intelligence and speech technology for a long time. Only recently, improvements in the technology of speech recognition, language understanding and dialogue modelling have made the implementation of spoken dialogue systems possible. Evaluation is essential for designing and developing successful spoken dialogue systems. The contributions of evaluation are to pin point weak or missing technologies as well as to measure the performance of individual system components, dialogue strategies and overall system performance.

AdApt is a research project for studying human-machine interaction in a multi-modal conversational dialogue system running at CTT (Centre for Speech Technology). The practical goal of AdApt is to build a multi-modal conversational system in which the user can collaborate with an animated agent to achieve complex tasks. The tasks of the system are associated with finding available apartments in Stockholm. A methodology in which system functionalities are simulated, Wizard of Oz, has earlier been used in the development of AdApt. Limitations of this methodology have led to a need for a tool which is capable of evaluating the system version available. AdApt is presently a system with fully functional components, simulation is no longer needed and a new method for evaluation is desirable.

Since spoken dialogue systems are created for the user, development of a spoken dialogue system should be in the light of the users' perception of the system. A tool for evaluation should try to reveal the users' perception of the system. Furthermore, to be capable of comparing different dialogue strategies and to guide designers in the development of spoken dialogue systems, evaluation should reveal whether changes made to the systems are perceived as improvements or not.

1.2 Purpose and method

This master's thesis presents experiences from an evaluation of AdApt, a multi-modal, conversational, dialogue system using PARADISE, PARAdigm for Dialogue System Evaluation. The purpose of the evaluation was to assess PARADISE as an evaluation tool for such a system. An experimental study with 26 subjects was performed. The subjects were asked to interact with one of three different system versions of AdApt. During the interaction with the system different types of data were collected.

PARADISE is a general framework for evaluation with the primary objective to maximize user satisfaction. The method combines various already acknowledged performance measures into a single performance evaluation function. PARADISE normalizes for task complexity and supports comparisons between different dialogue strategies and between spoken dialogue systems that carry out different tasks. The data in the study was collected through questionnaires, hand tagging and automatic logging of the dialogues.

A second purpose of the data collection was to make a comparison between three different versions of AdApt. The comparative study is described separately in Edlund and Nordstrand (2002), but in order to further test PARADISE as an evaluation tool, it was also applied to the three different system contributions. The system configurations tested were: (1) turn-taking gestures from an animated talking head, (2) an hourglass symbol to signal when the system was busy and (3) no turn-taking feedback at all. This was done in a between-subjects design with 8 subjects in each group

The report consists of a theoretical background (2) in which the issues of spoken dialogue systems and evaluation are discussed (2.1, 2.4). This section also includes detailed descriptions of PARADISE (2.5) and AdApt (2.6). The theoretical background is followed by a description of how PARADISE was applied to AdApt (3), a presentation of the results (4), and finally a discussion (5).

2 Theoretical background

Spoken language is a natural and efficient way for humans to communicate. We use language to buy milk in the supermarket, to gossip with our neighbour and to debate politics. Language use is a *joint action*, where the speaker and the listener perform their acts in coordination (Clark, 1997). Since language is frequently and efficiently used in a human-human dialogue it is close at hand to use in human-machine dialogue. Spoken language can hopefully result in a more natural and relaxed interaction between human and machine.

2.1 spoken dialogue systems

To create a conversational machine has been many scientists' dream even before the first computer was built. However, it is only recently spoken dialogue systems (SDS) have become a practical possibility. The implementation of these systems depends on progress made in the technology of speech recognition and understanding. The purpose of SDS is to provide an interface between a user and a computer application that permits spoken language interaction with the application, typically a database or an expert system.

The dialogues in the first generation of spoken dialogue systems were sometimes inefficient and unnatural. These systems made several important contributions to the areas of research and also led to a need for more sophisticated systems with more far reaching goals. Besides new and more challenging functionalities the major goals have been to create systems that are natural and easy to use.

2.1.1 Components of spoken dialogue systems

The term *dialogue systems* include a wide range of different systems (McTear, 2002). The system input can be either spoken, typed, or a combination of several different input channels. The output can be either spoken or written. Moreover, a dialogue system can be combined with visual output such as images, and tables. Interactive Voice Response Systems, IVR-systems, is a kind of dialogue system which only allow restricted input. To make the system "understand" what is asked for, the input has to be of a special kind and expressed in a certain way. Speech-based computer systems allow a more free interaction. These systems have no fixed input requirements. They engage in a conversation with the user rather than just responding to predetermined commands. The main components of a spoken language dialogue system are: speech recogniser, natural language analyser, dialogue manager, response generation and speech synthesizer.

Speech Recognition

It is a well-known fact that automatic speech recognition (ASR) is far from being as accurate and reliable as human recognisers (McTear, 2002). ASR for any spoken utterance, for a wide range of speakers in a noisy environment, is very difficult. The basic process of speech recognition involves finding a sequence of words, using a set of models, and matching these with the incoming user utterance. The speech signal, which is a continuous-time signal, is converted into discrete units. These units can be either units of sound, phonemes, or units of words. Speech recognition is difficult because of the complex nature of the speech signal. The same utterance has different acoustic realizations depending on linguistic, speaker and channel variability. The linguistic variability is caused by differences in intonation and co-articulation. Age, gender, mood and shape of the vocal

tract are factors that result in intra-speaker variability. The acoustic signal also depends on the transmission channel and background noise. Speech recognition in a spoken dialogue system has to challenge these obstacles and handle the following factors: speaker independence, vocabulary size, continuous speech and spontaneous conversational speech.

The complex task of the speech recognition component is to extract a sequence of words, which later can be computed by the language understanding component. Speaker independence is necessary if the system is designed to be used by a wide variety of users since it is very difficult to train the recogniser for every individual user (McTear, 2002). Therefore, speaker independent system needs to be trained with samples from a variety of speakers whose speech patterns is representative of the potential users. Speaker-independent recognition is less robust than speaker-dependent recognition. The size of the vocabulary varies between different applications. An application with only a few words in the vocabulary constrains the user while an application with a larger vocabulary is more flexible, but also more complex. Some spoken dialogue systems are designed to allow natural speech. Natural speech does not have any physical separation in the continuous-time speech signal which makes it difficult to decide the boundaries between the words. Natural speech is also spontaneous and unplanned. Spontaneous speech is characterized by disfluencies, false starts, pauses and fragments.

Some SDS are designed to handle interruptions, by allowing the user to “barge-in” over the system. Barge-ins are a possibility to speed up the dialogue for users who are familiar with the system. They can be useful to speed up the dialogue, but they can also degrade its quality. The interruption needs to be detected to reflect that a barge-in has occurred and the dialogue status has to be updated. A technical problem with barge-ins is that speech recognition can be affected by the echo from its own system response.

Language understanding

The theoretical foundations of the language-understanding component are linguistics, psychology and computational linguistics (McTear, 2002). The main purpose of the language-understanding component is to analyse the output from the speech recognition component and derive a meaning that can be computed by the dialogue manager. The output from the speech recogniser is not a single string of words; rather it is a set of ranked hypotheses. After they have been analysed by the language-understanding component; only a few of the hypotheses make sense. Language understanding involves both syntactic and semantic analysis. Syntactic analysis is done to determine the structure of a sequence of words from the speech recogniser. The semantic analysis, on the other hand, attempts to derive a meaning from the constituents. The design of the language understanding component depends on the nature of the output from the speech recogniser and the type of input required by the dialogue manager.

Dialogue Management

The dialogue manager is the central component of the spoken dialogue system (McTear, 2002). Its main purpose is to control the flow of the dialogue. This includes determining if the system has elicited adequate information from the user, contextual understanding, information retrieval and response generation. The dialogue manager tries to find out what information the user asks for, optionally consults an external application, such as a database, and finally reports the information back to the user. The processes are described in serial order, but typically they are not. It is difficult to determine what information the user is asking for since speech recognition is not perfect and many user utterances are ill formed. In many cases the system has to use verifications and

clarifications strategies in order to retrieve sufficient information. Various error-handling strategies are discussed in section 2.2.2.

Since the dialogue manager is the central component of a SDS, design decisions concerning the dialogue manager influence all other system components. Some systems are designed to support questions from the user at any time (Lamel, Rosset & Gauvain, 2000). Other systems restrict the vocabulary that can be accepted at particular points in the dialogue. Another important issue concerning the dialogue manager is how to robustly detect errors and recover from them.

External communication

Generally spoken dialogue systems require some kind of communication with an external source such as a database. This is necessary to retrieve the information that the user is asking for. For example: the data collected from an external source in a timetable information system contains information such as departure times, prices and destinations.

Response Generation

The response generation component composes a message that will be sent to the speech output component to be reported back to the user. The process includes deciding which information should be included, how the information should be structured and its syntactic structure. The response generation component can use simple pre-defined templates or complex natural language generation. Complex natural language generation has mainly been used in research prototype systems. A good guideline to follow is to only let the response generation component use words that can be processed by the recogniser, since users tend to mimic its behaviour (Skantze, 2002).

Speech output

The speech output component translates the output from the response generation component into spoken form. Some spoken dialogue systems use a simple template-filling mechanism with pre-recorded sound. This method is suitable for systems with fairly constant output. However, when a dialogue system “understands” more complex user input, more sophisticated ways of responding are needed. A better solution for more complex natural language systems with varying and unpredictable output is *text to speech synthesis* (TTS). Text to speech synthesis involves two tasks: text analysis and speech generation. Text analysis results in a linguistic representation, which the speech generation component uses to synthesize speech into waveform. Speech generation also involves generation of prosodic description such as rhythm and intonation.

2.1.2 Dialogue management strategies

Dialogue control is handled differently in different kinds of dialogue systems. The extent to which the two parties, the human and the machine, maintain the initiative in the dialogue differs. The control of the dialogue may be system-led, user-led or shared (*mixed initiative*). In a system-led dialogue the system asks the user a sequence of questions. In a user-led dialogue the user asks the questions and in a dialogue with mixed initiative control of the dialogue is shared. In a mixed-initiative dialogue the user is free to ask questions at any time, but the system can also ask questions to elicit missing pieces of information or to clarify unclear information. Design issues include: determining what questions the system should ask, in what order and when. McTear (2002) describes three main strategies for doing this: (1) finite-state (or graph), (2) frame and (3) agent based systems.

Finite state based systems

Finite state based systems use a system-led dialogue strategy. All questions in a finite state based system are predetermined. The strategy is particularly suitable to handle well-structured tasks and is often used in commercially available SDS. The user is normally restricted to use single words or short phrases as input. The input is therefore relatively easy to predict, which puts less technical demands on the speech recognition and the language understanding components. The advantage of finite-based systems is simplicity. Few errors of recognition and understanding lead to a comparatively high performance. Disadvantages are lack of flexibility and an unnatural dialogue. Finite-based systems are not suitable to model less well-structured tasks in which the dialogue order is difficult to predict.

Frame-based systems

Frame-based systems are also system-led, but they allow a limited degree of user initiative. Unlike finite-based systems frame-based systems do not use a pre-determined sequence of questions, they use predetermined slots or templates which are to be filled with information supplied by the users. The dialogue order is based on input from the user and what information the system is required to extract. The system will still maintain the initiative and ask questions, but the questions are not fixed in order. Natural language can be used as input for correcting errors of recognition and understanding. The number of dialogue turns and the transaction time for the dialogue can be reduced since the system is capable of handling natural language and multiple slot fillings. The dialogue flow may be more efficient and natural than in finite-based systems. However, the dialogue history that is used to determine the next system action is still fairly limited. Frame-based systems are not suitable for modelling more complex transactions.

Agent-based systems

The progress that has been made in the area of speech technology has led to more sophisticated systems with more far reaching goals. A finite-based or a frame-based system where each piece of information is asked for separately, possibly also with requests for confirmations, can result in extra dialogue turns. It would be far more efficient if the user could provide the system with several pieces of information in one single utterance, such as “I would like to have a two room apartment with a shower situated in Gamla Stan”. At the same time the system needs to be intuitive and easy to use. The benefits of a more natural dialogue is lost if the user constantly has to make corrections and ask for help. A natural spoken dialogue system has to be able to handle more complex utterances since there are many ways to express a particular request and even more ways to express an utterance in which several requests are combined.

The interaction between human and machine in agent-based system have many similarities with human-human dialogue. The user is free to use natural language and is not restricted to certain predetermined commands. Agent-based systems support more complex dialogues and are suitable for less well-structured tasks. The aim of these more advanced systems is to go from “recognising” words or commands to “understanding” the meaning of natural language. Agent-based systems have adopted techniques from artificial intelligence. These techniques are used to focus on collaboration between agents. A system that can “understand” and act upon fluently spoken language does not involve a single interaction; rather it involves a dialogue in which both the human and the machine contribute to the outcome. In dialogue where the initiative is shared between the user and the agent, the agent has to be able to reason over different models of the task and the current dialogue state. The interaction in agent-based systems is viewed as a conversation between two agents, where both

agents are capable of reasoning both about their own, and the other agent's beliefs and intentions. The interaction constantly depends on the present context and therefore the dialogue evolves in steps that build onto each other. Both agents cooperate to achieve a common goal. The user is free to introduce new topics and make contributions that are not constrained by earlier system prompts.

Disadvantages

Disadvantages of agent-based systems are that they are less robust and require more resources and more complex processing than finite-based and frame-based systems. The system must be capable of a deeper semantic representation to interpret the users' intentions. Collaborative problem solving requires more complex technologies such as techniques for clarifications and corrections. The complex dialogue in agent-based systems mainly affects three of the system components: the speech recogniser, which needs to handle a larger vocabulary, the language understanding component, which needs to parse the output from the speech recogniser and the dialogue manager, which has to cope with a number of different situations.

Naturalness

One of the most prominent goals of agent-based systems is to give the systems more intelligent and human-like behaviour. A dialogue can be understandable and usable but not human-like or "natural" (Boyce and Gorin, 1996). For example:

| | |
|---------|---|
| System: | Please say your authorization code now. |
| User: | 5 1 2 3 4 |
| System: | Invalid entry. Please repeat. |
| User: | 5 1 2 3 4 |

This dialogue is perfectly understandable but to make it natural you would have to use elements from human-human dialogue. Learning how to pose complex database queries can be difficult for novice users. A system that supports natural language enables the users to use information he or she already has about dialogues it in their interaction with the system. The users can consequently rely on what they already know about language and conversation.

2.2 Errors in human-machine dialogue

Face-to-face conversation between humans is not without problems despite the fact that this is a skill we have been developing since we were born. Given the difficulties that occur in human-human dialogue it is naïve to believe that human-machine dialogue is easy. For most users this is a whole new situation. No time has been spent on establishing a consistent *common ground* (Clark, 1997). The common ground of a system and a user is the sum of their mutual knowledge, beliefs and suppositions. According to Clark communication is impossible without common ground. The initial common ground between a system and a user includes the task domain and the modalities that can be used for interacting. If SDS includes a GUI (graphical user interface) this can be an important additional contribution, which helps the user and the system to further establish their common ground. Still, this common ground might not be enough for the user and the system to engage in a rewarding dialogue. Human-machine dialogue is complex and difficult. As have been discussed earlier: one guiding principle when developing SDS is to use elements from human-human dialogue. However, computers are not humans and computers do not have the full power of human conversational competence. Moreover, humans behave differently when they are talking to a machine rather than to a human.

2.2.1 *Setting expectations*

It is essential that the system accurately conveys the system functionalities to the user. The users tend to believe that the system has greater capabilities than it actually has when using a system for the first time (Litman & Pan, 1999). The system greeting is particularly crucial because it sets the scene for the rest of the dialogue. To set the right expectations is one of the more difficult issues in the design of SDS. This includes conveying both what kind of speech input is required and the knowledge domain that the system can process. Novice users may have major difficulties if the expectations are not well matched with the capabilities of the system. This can result in a bad system performance since user utterances are more likely to be rejected or misunderstood. Furthermore, dialogue length typically increases and users are less likely to achieve their task goals.

A solution to the problem of setting the correct expectations is to lead the user through the dialogue using series of questions (system-led). The questions will clearly illustrate the capabilities of the system. Unfortunately, this results in a less flexible and less natural system, which restricts the users. On the other hand, providing the user with more freedom sometimes brings difficulties for the user to understand the scope of the domain.

2.2.2 *Misunderstandings and non-understanding*

In human-human communication utterances are frequently misheard, misunderstood or entirely missed. Although humans are superior recognisers, human-human dialogue is not free from errors of recognition and understanding. Detecting miscommunications and repairing them by initialising appropriate repair sub dialogues is essential in human-human dialogue. Detection and correction of errors are therefore also major design issues in the development of SDS. However, error management also increases the complexity of the dialogue and can lead to problematic dialogue management problems (Turunen & Hakulinen, 2001). According to McGee, Cohen and Oviatt (1998) humans try to avoid misunderstandings in two ways: (1) by acknowledging what others are saying and (2) by requesting confirmation when there is doubt of what was said.

The dialogue manager occasionally comes up with several likely interpretations of a user utterance and needs to clarify which interpretation is most prone to be correct. There are a number of different clarifying dialogue strategies (Boyce & Gorin, 1996). For example: “Do you want A or B?”. Another strategy is to ask a yes/no question such as “Do you want A?”. If the answer is “No” the system can choose to presume B. The choice of strategy depends on the relative confidences of the generated interpretations. A difficult issue is for the system to determine when an utterance is “misunderstood”, i.e. when confidence is too low. The system has to determine whether the possible interpretations are likely or not. Non-understandings are less complicated to handle since it is fairly easy to detect when no input at all was received.

Misunderstandings

Failures of recognition and understanding can cause actual human-machine misunderstanding. Verification strategies are used to deal with potentially misrecognised input. The system is “aware” of that the input may have been misunderstood or misrecognised and needs to clarify what was actually said. In human-human dialogue confirmations are used to make sure that what was said is mutually understood and to establish a common ground (Clark, 1997). Confirmation and verification strategies are two of the most challenging issues in the design of spoken dialogue systems.

Explicit verifications

There are several different ways to verify that a user’s utterance has been correctly understood. Explicit verifications are openly stated requests for confirmation of the user input. Explicit verifications may be accompanied by a request to answer yes or no. For example:

“Do you need X? Yes or no?”

In these openly stated verifications two values are confirmed at the same time. An advantage of explicit confirmations is that they are very straightforward and the user knows immediately how to respond. However, it is difficult if both values are incorrect. One solution to this problem is to confirm each value separately:

“Do you need X?”

“Do you need Y?”

This strategy is easy to know how to act upon, but it will increase the total number of utterances in the dialogue.

Implicit Verifications

Implicit verifications are more frequently used in human-human dialogue than explicit verifications. Implicit verifications can be more effective and are a less openly stated confirmation strategy:

“Ok, you need X”.

One alternative is to embed the next question in the implicit verification.

“Ok, you need X. Do you need Y?”

This is an even more effective confirmation strategy since the system combines the verification with a question which takes the dialogue one step further. Nevertheless, there is still a possibility for the user to correct the system, but if the question is being answered without a correction the value is implicitly confirmed. The problem with implicit confirmations is that they result in a wider range of possible user responses, which puts greater demands on the system in terms of speech recognition (McTear, 2002). Furthermore, the language-understanding processes are more complex. The benefits of implicit confirmations are a more natural and effective dialogue. The negative aspects of implicit confirmations are errors of recognition and the fact that they are less straightforward to correct.

Experimental results suggest that confirmations are even more essential in human-machine interaction than in human-human interaction (Boyce & Gorin, 1996). In a study by Boyce and Gorin implicit and explicit confirmation strategies were studied. Both strategies turned out to be very successful when the system made correct interpretations of the users' utterances. However, when the interpretation was incorrect the users with the implicit confirmation strategy were less successful in repairing errors. The data suggests that explicit confirmation is a more robust method. However, this does not necessarily mean that the explicit verification strategy is superior to the implicit verification strategy, since Boyce and Gorin did not test how the different confirmation strategies were perceived by the users. A more natural system with implicit confirmations might have more positive effect than robustness on user satisfaction and result in a more positive perception of the system. Implicit confirmations may result in an increased number of misunderstandings while explicit confirmations will result in an unreasonable number of dialogue turns.

Non-understanding

Non-understanding occurs when confidence is too low for the system to come up with any possible interpretation. Consequently, the system completely fails to interpret the user's utterance. The failure may have occurred either as a result of a speech recognition error or because the language-understanding component was not able to interpret it. Non-understanding is a less problematic miscommunication than misunderstanding since it is usually recognized by the system as soon as it occurs. A breakdown is typically handled with a repair action, typically a request for the user to rephrase the utterance. These requests for repetition are called *reprompts* (Boyce and Gorin, 1996). Reprompts are also used when the confidence is very low and it is better to ask for repetition. A request for verification of an interpretation that is almost certainly wrong is not a good solution. The simplest way of dealing with ill-formed and incomplete input is to simply ask for repetition: "Please repeat". This method is inadequate since it does not provide any information that reveals in what way the input was incomplete or ill formed. It fails to support the user in reformulating the input. Humans have a broad spectrum of strategies to use when an utterance is not understood. Which strategy is chosen depends on what went wrong. The utterance could, for example, have been either misunderstood or misheard. When an utterance is only partially understood, additional information is needed. The different strategies humans normally use are: reprompts, clarifications and to silently wait for more information. If the system could intelligently imitate these features of human-human dialogue it would also be able to illustrate which part of the dialogue went wrong. If the user knows what part of the interaction went wrong, correction of errors will be less problematic. If the dialogue has suffered from repeated breakdowns the user should be brought in to complete the transaction. This should preferably occur before the users' frustration gets to big and results in a negative perception of the system.

Reprompts were studied in an experiment by Boyce and Gorin (1996). The reprompts in the study fell into two categories. (1) An apology followed by a restatement of the original prompt:

“I’m sorry. How may I help you?”

(2) The other category of reprompts included an explicit statement which declared that the utterance was not understood:

“I’m sorry your response was not understood. Please tell me again how I can help you?”

The results showed no major difference between the two categories of reprompts. The users responded to reprompts differently; they either repeated what they just said or chose to rephrase themselves. Which strategy was adopted can depend on the user’s mental model of the system failure (Boyce & Gorin, 1996). When repeating an utterance humans tend to over-articulate and speak slower. This leads to a degradation of the system performance. One third of the times the users in the test interpreted the reprompt as a request for them to repeat the same utterance. This is not always a desirable solution in terms of system performance since the speech recogniser will probably not do better a second time. 80% of the subjects who gave a shorter response the second time included the same amount of information, only rephrased shorter. The other 20% shortened their response by giving less information. This could reflect a belief that the first utterance contained too much information and that the system only is able to handle one chunk of information at a time. The results from the test imply that the users adopted different strategies when replying to reprompts. According to Boyce and Gorin (1996) this could be the result of different mental models of system failures.

2.3 Multi-modal Spoken Dialogue Systems

Clark (1997) uses the term *signal* for any action by one person that is intended to mean something to another person. A signal is a deliberate action initiated by one person for another person to identify. According to Clark a signal includes all actions that fulfil these criteria. To open the door for someone, to shake hands or to rise an eyebrow at someone are all different kinds of signals. Consequently we have more ways to communicate with other humans than speech. A multi-modal system is a one that supports communication with users through different modalities such as voice, typing and gesture.

Speech-only interfaces have shown a number of shortcomings that result in inefficient dialogues. Adding extra input/output modalities may be a way to reduce some of the problems in human-machine dialogue. A new generation of computer interfaces has the ability to interpret combinations of multi-modal input and to generate a coordinated multi-modal output. These interfaces can benefit in efficiency and naturalness since combinations of modalities can help to reinforce and disambiguate each other. However, multi-modal applications are complex and require expertise from different technologies, academic disciplines and cultural perspectives (Oviatt, 2000).

2.3.1 Multi-modality

Humans have several modalities, perception channels, for example visual, audible and tactile. The modalities are used to send and retrieve information. When more than one modality is involved we speak of *multi-modality*. Humans interacting with the outer world use a combination of their modalities. It is therefore a natural solution to provide machines with the ability to interact on the same terms. The opinion whether a multi-modal system has to be able to handle both multi-modal input and output differs.

One of the shortcomings of speech-only interfaces is the imperfection of speech recognisers (Sturm, Wang & Cranen, 2001). Requests for repetition often lead users to over-articulate, which most likely will result in an even worse performance. Another design issue is how to design a robust and effective confirmation strategy for a system with speech as the only input modality. Explicit confirmations result in an inefficient dialogue with extra turns while users have difficulties to grasp the concept of implicit confirmations. Finally, users also appear to have difficulties to build a correct mental model of the functionality of the system. A solution of these problems can be multi-modality (Sturm, Wang & Cranen, 2001). Adding an extra output modality may give the user a better mental-model of the system and adding an extra input modality may result in better interpretations of the users' utterances.

2.3.2 Speech and Gesture

Speech and gesture is the most common modality combination in human-human dialogue. For this reason the combination seems rewarding to use in human-machine dialogue. Multi-modality is not an attempt to allow several modalities to cohabit; rather it is an attempt to allow different modalities to cooperate. For instance, the user might use speech to inform the system how to manipulate an object while using a gesture to select which object to manipulate. From a linguistic perspective speech and gestures are often viewed as modalities that carry different semantic content. According to Oviatt (1997): "gesture has been viewed as a cognitive aid in the realization of thinking". In human-human dialogue the modalities are naturally synchronized. The new generation of multi-modal systems tries to imitate the synchronized whole of different modalities that characterizes human-human communication.

Multi-modal input

Spoken dialogue systems have used speech as the only input modality for a long time. Speech is often considered to be the most natural input since it is the primary means of human-human communication (Sturm, Wang & Cranen, 2001). The advantage of speech only interfaces is that except for a microphone, they do not require any additional devices. Furthermore, they are superior when both hands and eyes are busy. However, speech-only interfaces have, as have been discussed earlier, shown a number of shortcomings.

Coordination of modalities is one of the most important design issues in the development of multi-modal SDS. Synchronization in human-human dialogue comes naturally. Unfortunately the coordination of modalities in human-machine dialogue is much more complex (Oviatt, De Angeli, & Kuhn, 1997). For example: the speech recogniser component often needs more time to register an utterance than the device for gestures needs for computing the coordinates of a mouse-click or a pointing gesture on a touch-screen. Lack of coordination between different components can lead to interpretation difficulties (Bellik, 1994). What criteria should be used to decide whether input from two different modalities should be interpreted in combination or separately? Time is an important factor. Another central factor is the technical constraints of the components for the different modalities. For instance operations that require high security should be assigned to the modalities that have few errors of recognition.

2.3.3 Animated synthetic faces

Information transmitted through the optic channel in human-human interaction is generally underestimated (Benoît, 1992). It has been a well-known fact for some time that hearing-impaired depend to a great extent on lip-reading to understand speech, but also normal hearing humans have an element of visual hearing. Normal hearers seem to rely primarily on the auditory modality in speech, but if visual information is available this can facilitate the progress of speech understanding. This is especially prominent in a noisy environment. However, it is difficult to intelligibly perceive speech from the visual modality alone.

In a face-to-face conversation speech is transmitted both audibly and visually (Benoît, 1992). Continuous speech is made up of pauses of silence where the speaker makes gestures in order to anticipate the following sound. The visual parts of the sound, our lips, tongue and jaw also convey useful information. To sum up, speech is a combination of some parts that are only audible, some parts that are only visible and some parts that are both visible and audible.

Visible speech is particularly effective when the auditory modality is degraded for some reason. The degradation can be due to hearing-impairments, environmental noise or bandwidth filtering (Benoît, 1992). When presented with combined modalities, visual and auditory, performance jumps remarkably. Facial gestures can therefore be useful in a SDS with synthesized speech, since synthesized speech may be perceived as unnatural and sometimes is difficult to comprehend by novice users. Facial gestures can help to reduce these effects. However, it is important that the visible gestures are well synchronized with the audible speech or else the gestures might have a reverse effect. The *McGurk-effect* is an example of this. McGurk showed that a simultaneous presentation of an acoustic “ba” and a visual “ga” was perceived by the listeners/viewers as “da”.

Animated humans and animal like agents have recently been applied to HCI (Human Computer Interaction). However, many of these depictions have been more decorative than helpful (Cassell, 2000). Some agents seem to be designed to bring a pleasant novelty to the system rather than to actually make meaningful contributions to the dialogue. Human-human dialogue is characterized by face-to-face conversation and being able to realistically mimic these features in human-machine dialogue would be rewarding. Some features from face-to-face conversation that could fruitfully be applied to human-machine include: mixed initiative, non-verbal communication, sense of presence and rules for transfer of control. According to Cassell et al. (1999): “interfaces that are truly conversational have the promise of being more intuitive to learn, more resistant to communication breakdown, and more functional in noisy environments” Furthermore, they suggest that implementing as many as possible of the communicative skills that humans have will bring dialogue systems closer to being “truly conversational”.

A variety of synthetic faces have been developed during the last two decades. An animated face can be an important complement to speech synthesis (Bickmore & Cassell, 2001). The display of a synthetic face consistently animated in synchrony with synthetic speech makes the synthesizer sound more pleasant and natural. Furthermore, a face has the capacity to express emotion, add emphasis to speech and assist the dialogue with turn-taking gestures and back channelling (Beskow, 1995). Important information can be expressed by rising and shaping the eyebrows, eye movements and nodding of the head. Head movements are used to put emphasis on speech but also to make the face look more alive. However, these movements can be difficult to model since they are depending on mood and personality of the speaker. Yet, typically humans tend to raise their eyebrows at the end of a question and at a stressed syllable. An animated face can also help to build trust with the user.

2.4 Evaluation

Evaluation is essential for designing and developing successful spoken dialogue systems. There is a long tradition of quantitative performance evaluation in information retrieval and many of its concepts have been adopted to the development of evaluation methodologies for speech and natural language processing. The contributions of evaluation are to pinpoint weak or missing functionalities and to evaluate the performance of individual components or utterances as well as the overall system performance. Peak (2001) suggests four typical bases for evaluation:

- (1) Provide an accurate estimation of how well a system meets the goals of the domain task.
- (2) Allow for comparative judgments of one system against another, and if possible, across different domain tasks.
- (3) Identify factors or components in the system that can be improved.
- (4) Discover tradeoffs or correlations between factors.

The study of human-human dialogue provides some useful insights into the nature of the interaction in SDS, but it is somehow limited since humans behave differently when talking to a machine rather than to another human. This is in line with Clark's joint action theory (1996). Language use is a joint action and the dialogue is depending on both the speaker and the listener. The computer plays either the role of the listener or the speaker in a SDS and its characteristics will have impact on the dialogue. Therefore; corpora from human-machine dialogue can be very useful in the development of SDS. There are, however, a number of different methodologies and metrics available for collecting corpora.

2.4.1 *Wizard of Oz*

Developing conversational interfaces is a chicken and egg problem. To create a working system a large corpus of data for system development is needed for training and evaluation. To collect data one needs a working system. *Wizard of Oz* methodology is useful for making evaluations in the early stages of system development before significant resources have been invested in system building (Giachin, 1995). In this method, a human simulates the role of the computer. The user is made to believe that he or she is interacting with a machine through synthesized speech, possibly combined with a graphical user interface (GUI). In reality the speech recognition and natural language understanding components are simulated by a human. The experimenter is in charge of the dialogue and can use controlled scenarios. The Wizard of Oz methodology has been useful for researchers to test ideas. It is for example possible to simulate errors, which is useful for testing error recovery strategies. However, Wizard of Oz is problematic, since it is very difficult to realistically mimic the behaviour of the system. The method can therefore result in a dialogue strategy which is not robust when used in a real, future version of the system.

2.4.2 *System in the Loop*

To overcome the problems of how to realistically mimic the behaviour of the system with Wizard of Oz methodology, *System in the Loop* may be used (McTear, 2002). This is a different approach, where the system version that is available is tested, and no functionalities are simulated. Even a system with limited functionality can be useful for collecting data. The idea behind system in the loop is that additional functions can be implemented later on to be evaluated in the next cycle of testing. Unlike Wizard of Oz methodology this approach will not lead to false imitations of the system's behaviour. However, system in the loop can be difficult to use in the earlier stages when there are no, or only a

few, system functionalities implemented to test. Wizard of Oz and System in the Loop are not mutually exclusive, but can be used in combination. Whether the system performance is simulated or real, collected corpora from human-machine interaction are playing an essential role in spoken dialogue system development.

2.4.3 Objective metrics

Evaluation of spoken dialogue systems is difficult, since spoken dialogue is complex and ill defined. Performance evaluation of spoken dialogue system components has successfully been used (e.g. DARPA) to assess various functionalities: robust information extraction from text, large vocabulary continuous speech recognition and large-scale information retrieval. These evaluations have motivated researchers, both to compete in building advanced systems and to share information in order to solve the problems. Amongst the contributions of these evaluations are increased communication among researchers, increased visibility for the research areas in question, and rapid technical progress. However, the focus of attention has been on the underlying technologies, rather than on complete applications. Designers have often used *objective metrics*. Objective metrics are automatically logged and automatically calculated, and are therefore easy to apply (Antoine, Siroux, Caelen, Villaneau, Goulian, & Ahafhaf, 2000). Secondly, objective metrics can be calculated automatically and do not require human evaluators to make them reliable. Furthermore, objective metrics can easily be compared, since they are quantitative.

Commonly used objective metrics include number of words and utterances per task as well as task success based on *reference answers*. They are used as pre-defined desired output to be compared to the actual output (Walker, Litman, Kamm, & Abella, 1998). Reference answers are relatively easy to determine for speech recogniser and language understanding components, but can be difficult to determine for the dialogue manager, since the range of acceptable behaviours is much greater. Another acknowledged limitation is that the use of reference answers makes it difficult, if not impossible, to compare systems that carry out different tasks. The reason is that one correct response needs to be defined for every user utterance, even though there might be a large number of possible correct answers. For example: it is not possible to compare a system that gives a list of database values as a response to a system that gives an abstract summary as a response. Objective metrics are, in general, not suitable for comparing different tasks, since the task complexity varies.

2.4.4 Subjective metrics

Designers face a number of complicated issues when evaluating spoken dialogue systems. There has been great progress in developing objective evaluation metrics for individual components, but the overall quality of a software system is not solely depending on the functionality and performance of its constituents (Bernsen & Dybkjaer, 2000). A fast system with a high success rate does not necessarily please the users. Furthermore, objective measures are, from the users' point of view, not suitable for evaluating the overall system quality. The system quality is constantly changing depending on surrounding factors such as the user, the type of task to be carried out and other available alternatives. A system can be technically superior, which is an objective quality, but the users does not necessarily perceive it as superior.

The overall performance of a spoken dialogue system depends on how the different components perform as a whole, as well as on how the users perceive the system. The performance of a particular system component is affected by the performance of the other components and, conversely, its performance might degrade or improve the performance of the other components. SDS are

ultimately created for the users and the obvious goal is to maximise their satisfaction (Giachin, 1995). User satisfaction is generally based on *subjective metrics*, which are collected through user surveys. Subjective metrics require subjects using the system and human evaluators to categorise sub-dialogues or utterances within the dialogue along various qualitative dimensions. Subjective metrics can still be quantitative, as when the number of occurrences of a certain qualitative category is calculated. Subjective metrics may be used to determine what parameters are critical to performance. If the critical aspects of a system are known, less important technical aspects of the system may be ignored. Since usability factors are based on human judgment they need to be reliable across judges. Subjective metrics can be qualitative in the shape of user surveys. These user surveys can be questionnaires, which typically include questions about naturalness, clarity, friendliness, subjective length, and error handling. The subjective metrics can also be quantitative. Examples of quantitative metrics that can be subjective are for example percentage of implicit or explicit recovery utterances. Subjective measures are useful for detecting weak points and neglecting less important technical aspects of the system.

2.4.5 A general framework for evaluation

Interactive spoken dialogue systems consist of several different components. The overall functionality of a system depends on the overall performance of these components. For example: a good performing language-understanding component might compensate for a bad performing speech recogniser. Consequently, using individual component metrics can be misleading since they do not reveal how the different system components contribute to the overall performance. What criteria should be used to determine the overall performance of a SDS? SDS are ultimately created for the users and the obvious goal is to maximize user satisfaction. However, user satisfaction is a subjective goal and can highly depend on individual differences between users. A second acknowledged limitation is that individual component evaluations make it difficult to combine various metrics and to make generalizations. An example of this is a comparison between two different timetable information agents, A and B, where A and B use different dialogue strategies (Danieli & Gerbino, 1995). Agent A uses an explicit confirmation strategy in dialogue 1 and agent B uses an implicit strategy in dialogue 2:

- (1) User: I want to go from Torino to Milano.
Agent A: Do you want to go from Trento to Milano? Yes or no?
User: No

- (2) User: I want to travel from Torino to Milano.
Agent B: At which time do you want to leave form Merano to Milano?
User: No, I want to leave from Torino in the evening.

Results from an evaluation of these two dialogue strategies showed that the explicit dialogue strategy (Agent A) had a higher transaction success rate and fewer repair utterances than the implicit dialogue strategy (Agent B). However, the dialogues with the implicit dialogue strategy (Agent B) were about half as long as those with the explicit dialogue strategy. Because of the inability to combine the metrics it was impossible to determine whether a high transaction success rate with few repair utterances or an efficient dialogue was most critical to performance. Danieli and Gerbino (1995) suggest that an important research topic is definition of methods for evaluating the systems' effectiveness and friendliness from the users' point of view. A method which identifies how different multiple factors affect the overall performance is also necessary if one wants to make generalizations across different systems performing different tasks. According to Walker, Litman, Kamm and Abella

(1998): “It would be useful to know how users’ perceptions of performance depend on the strategy used, and on tradeoffs among factors like efficiency, usability and accuracy”. Occasionally, different metrics contradict each other (Peak, 2001). Contradicting metrics leaves the designer with the tricky task of finding interactions and correlations between the metrics.

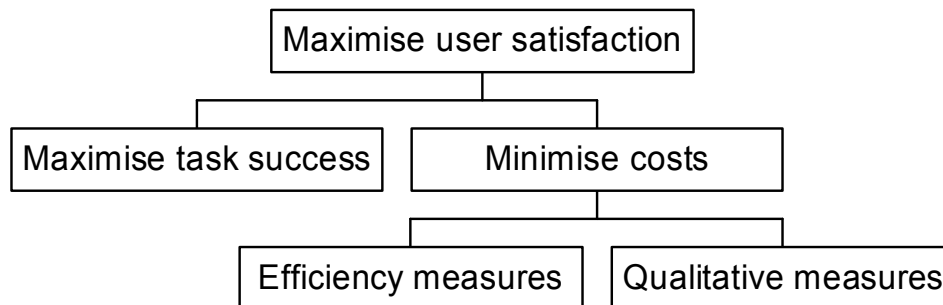
2.4.6 Evaluation of multi-modal systems

The goal of multi-modal spoken dialogue systems are the same as for SDS which use only one modality; they should be efficient, intuitive and above all, useable. Since multi-modal systems are more complex, so will also the evaluation of these systems be. There are several challenges in designing evaluation methods for multi-modal dialogue systems. These evaluations require more than one perspective of testing and more methods of logging.

2.5 PARADISE

The need for a general evaluation method came with the problems from using several different metrics (Walker et. al, 1998). Peak (2001) suggests that: “instead of focusing on developing a new metric that circumvents the problems described earlier, the designers need to make better use of the ones that already exist”. PARADISE (PARAdigm for Dialogue System Evaluation) is a general framework for evaluating spoken dialogue systems that combines various already acknowledged performance measures into a single performance evaluation function.

PARADISE uses multivariate linear regression to combine a set of different performance metrics and specify how these multiple factors contribute to the overall performance. Comparisons between different dialogue strategies are supported by a task representation that decouples the goal of the task from how it is carried out. The ability to illustrate how different components contribute to the overall performance is also used to make comparisons between systems performing different tasks. Another benefit from using PARADISE is that performance can be calculated on whole dialogues as well as on sub-dialogues (Walker et. al, 1998). To calculate the performance of sub-dialogues you would have to assume that the factors that contribute to the global performance are generalisable. Factors that are assumed to be generalisable can also be used to predict the performance of sub-dialogues. The PARADISE model argues that the overall goal of a spoken dialogue agent is to maximize user satisfaction:



The model further posits that the top-level objective, user satisfaction, depends on two potential contributors: task success and dialogue costs (Walker, Litman, Kamm & Abella, 1998). Dialogue efficiency and dialogue quality are, in their turn, potential contributors to dialogue costs. The efficiency measures are intended to illustrate how efficient the system is in helping the user to complete a task, i.e. how long or how many utterances it takes to complete a task. The purpose of the qualitative measures is to capture the features of the system that influence the users’ subjective ratings.

2.5.1 Task definition in PARADISE

A general framework for system evaluation has to decouple *what* is accomplished from *how* it is being done (Walker, Litman, Kamm & Abella, 1998). This is necessary if you want to compare different dialogue strategies. PARADISE represents tasks in Attribute Value Matrixes (AVMs). An AVM consists of the information that needs to be exchanged between a user and an agent during a dialogue to perform a certain task scenario. The information is represented as ordered pairs of attributes and their possible values.

A simple AVM for the following dialogue looks like this:

- 1.R-1 Östermalm.
Östermalm.
- 1.S-1 Hur många rum vill du ha?
How many rooms do you want?
- 1.R-2 Två rum.
Two rooms.
- 1.S-2 Hur mycket får lägenheten kosta?
How much can the apartment cost?
- 1.R-3 Tre miljoner.
Three millions.
- 1.S-3 Det finns 7 sådana lägenheter och de visas nu på kartan.
There are seven such apartments and they are displayed on the map.

R - User utterances registered by the system

S - System utterances

The utterances are numbered to facilitate reference.

| Attribute | Actual value |
|------------|--------------|
| Area | Östermalm |
| Size | 2 rooms |
| Price | 3000000 Skr |
| Apartments | 7 apartments |

Figure 2.1: Attribute Value Matrix

In this dialogue the agent needs to obtain values for area, size and price from the user. The user, on the other hand, needs to obtain the apartment value from the agent. The apartment value provides the user with information about which apartments in the database fulfil the criteria of the values that he/she has already given. Note that two different agents with different dialogue strategies who carry out the same task have the same attribute value matrix. AVMs represent the attribute value pairs as they exist at the end of the dialogue.

2.5.2 Measuring Task Success

The overall performance function in PARADISE requires dialogue corpora to be collected through a set of predetermined user scenarios (Walker, Litman, Kamm & Abella, 1998). Each scenario has a corresponding AVM that represents the information that was actually exchanged in the dialogue. How well the system has succeeded in carrying out a certain task in a dialogue, or sub dialogue, is a definition of how well the system and the user has succeeded in carrying out the information requirements when the dialogue or sub dialogue is completed. All dialogues that result in an AVM, which corresponds to the AVM that instantiates the information requirements of the task, are considered to be successful. Task success is therefore independent of the dialogue structure. Long and inefficient dialogues with many repair utterances are reflected in the dialogue costs. The matrixes

can also be increased to handle several correct answers; the attributes in the matrix can consequently have more than one value. According to Walker et al. (1998), one way to do this is to represent the values as disjunctions of possible values. Another solution is to redesign the AVMs as entity relationship models (Korth, 1997).

The Kappa coefficient

The Kappa coefficient in PARADISE is used to operationalize the task-based success measure. The Kappa coefficient is calculated on a *confusion matrix* (see appendix 8.1). A confusion matrix is a summary of how well the agent and the user have succeeded in accomplishing the information requirements for a specific task, instantiating a set of task scenarios. The values in the cells of the confusion matrix are based on a comparison between the actual value that is exchanged in the dialogue and the scenario key in the AVM. Whenever a dialogue value matches the scenario key the number in the appropriate diagonal cell of the matrix is increased by one. Misunderstandings, which have not been corrected during the dialogue, are represented in the off-diagonal cells. The misunderstandings that are corrected during the dialogue are not left unnoticed. The time spent on the dialogue and the number of utterances is reflected in the dialogue costs. How well the agent has succeeded in conveying the information requirements of the task is measured using a *kappa coefficient*:

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

(1) The Kappa Coefficient

$P(A)$ is the number of times that the AVMs for the actual set of dialogues agree with the AVMs for the scenario keys. $P(A)$ is computed given a confusion matrix M :

$$P(A) = \frac{\sum_{i=1}^n M(i, i)}{T}$$

(3) $P(A)$

$P(E)$ is the number of times that the actual values and the scenario keys are expected to agree by chance. When the prior distribution of the categories is unknown, the expected chance agreement between the data and the key, $P(E)$, can be estimated through:

$$P(E) = \sum_{i=1}^n \left(\frac{t_i}{T} \right)^2$$

(4) $P(E)$

t_i is the sum of the frequencies in a column of the confusion matrix, and T is the sum of the frequencies of all the columns in the confusion matrix, $M(t_1 + \dots + t_n)$. When there is no agreement at all, except for the agreement that are to expected to agree by chance, $K=0$. For total agreement $K=1$. According to the authors of PARADISE: the Kappa coefficient is superior to other task based success measures such as concept accuracy, percent agreement and transaction success. The Kappa coefficients superiority is due to its ability to normalize for task complexity and make comparisons between different agents performing different tasks. For further details see (Walker, Litman, Kamm and Abella, 1997).

2.5.3 Dialogue Costs

According to PARADISE the system's performance is a combination of task success and dialogue costs. Instinctively, the dialogue costs should correspond to the user or agent behaviour that should be changed or reduced. As have been discussed earlier, there are a number of different subjective and objective metrics available for evaluating SDS. Since it is impossible to know in advance what factors will contribute to usability it is necessary to use several different metrics. Furthermore, the same set of metrics has to be used to be able to make generalizations across different tasks. The cost based measures in PARADISE are represented as C_i . C_i can be applied to any dialogue. To calculate the dialogue costs for sub-dialogues and for some of the qualitative metrics it is necessary to specify what information requirements a certain utterance contributes to. The AVMs link the information requirements of a task to arbitrary dialogue behaviour through attribute tagging. The dialogue is labelled with the task attributes. The labelling makes it possible to evaluate potential dialogue strategies for the whole dialogue as well as evaluating dialogues strategies for sub-tasks in sub-dialogues. It is necessary to label the attributes in the AVM to be able to calculate dialogue costs over sub-dialogues since a sub-dialogue is defined by the attributes of the task. Furthermore, the labelling is necessary to calculate the costs for some qualitative measures, such as the number of repair utterances. For a given C_i the different dialogue costs measures have to be combined to calculate for their relative contribution to the overall performance.

2.5.4 The Performance Function

The weights in the performance function are calculated through correlating the users' subjective judgments with the system's performance (dialogue costs and task success) using multivariate linear regression. The performance function can be used to predict future versions of the agent. It can also be used as a foundation for feedback so the agent can learn to optimise its behaviour based on its experiences with users over time. The task success and the dialogue costs are used to create the overall performance function:

$$Performance = (\alpha * N(\kappa)) - \sum_{i=1}^n \omega_i * N(c_i)$$

- α - is a weight on the Kappa coefficient
- (K) - is the Kappa coefficient
- ω_i - is weigth on the dialogue costs
- C_i - is the dialogue costs
- N - is the Z score normalisation

(4) The overall performance function

α and ω_i are calculated in the multivariate linear regression, i.e. the coefficients of the statistically significant predictors of User satisfaction. N is used as a Z score normalization function to overcome the problems of C_i and K not being on the same scales and the fact that C_i can be calculated over varying scales. If C_i and K are not normalised, the magnitudes of these values will not reflect the relative contribution of each factor to performance. Each factor, x , is therefore normalized to its Z score:

$$N(x) = \frac{X - \bar{X}}{\sigma x}$$

The predictive performance function in PARADISE makes it easier to do repeated evaluations. Once the weights in the performance function have been solved for, no more user satisfaction ratings need to be collected since predictions of user satisfaction can be made using the predictor variables. To be able to do this, models have to be generalisable across user populations and other systems.

2.6 AdApt a multi-modal conversational dialogue system

AdApt is a multi-modal conversational dialogue system developed at CTT (Centre for Speech Technology) at the Royal Institute of Technology in Stockholm with Telia Research as an industrial partner. The aim of the project is to study human-computer interaction in a multi-modal conversational dialogue system. The practical goal of AdApt is to build a multi-modal conversational system in which the user can collaborate with an animated agent to achieve complex tasks. The tasks of the system are associated with finding available apartments in Stockholm. The apartment domain was chosen for several reasons: the complex nature of apartments makes the domain suitable for multi-modal interaction; the locations can be referred to graphically and prices and interior properties verbally. The domain is known to engage a wide variety of people, i.e. possible users, living in the Stockholm area. Furthermore, the domain attracts people regardless of whether they are seriously thinking of purchasing a new apartment. The apartment domain, particularly in Stockholm, is one that many people want to keep up to date with to see what objects are available where at what prices.

2.6.1 Architecture

The AdApt system has all the functional components of a spoken dialogue system: a speech recogniser, a natural language analyser, a speech synthesizer and a dialogue manager. The system also has components for registering and analysing mouse input. The mouse can be used to select certain objects by clicking on them or marking areas on an interactive map of Stockholm. The output of AdApt is also multi-modal. Except for speech, the system output includes a visual map and a 3D-animated agent (Beskow, 1997). The animated agent produces lip-synchronized synthetic speech and gestures, and the apartments are displayed on the map as coloured dots. One important aspect of the multi-modality of AdApt is that the result from the speech input and the mouse are integrated. The dialogue manager consequently has to handle integrating in- and output of the different modalities, resolving anaphor and contextual references. Furthermore, it has to be capable of understanding the pragmatics of the input utterances as well as handling misunderstandings and non-understandings in a robust and efficient way. Coordination of audiovisual synthesis is another important issue. The database used by AdApt is extracted from real advertisements the web and contains information about size, price, location and apartment attributes such as balcony, bathtub and parquet.

2.6.2 Research goals

The AdApt project addresses a number of research issues. The long-term research areas are (Gustavsson et. al, 2000):

- 1) How do people make use of the different modalities and what are the implications of their choices in terms of system architecture?
- 2) How should the system interpret references, not only to objects previously mentioned in the dialogue, but also to objects currently visible on the screen?
- 3) Can multi-modality be used to increase the robustness of a spoken dialogue system? Users often change from one modality to another when the human-computer interaction becomes difficult.
- 4) How does the multi-modal setting with a mouse, an interactive map and an animated speaking agent influence how people speak?
- 5) Can the system influence users in their choice of modality?

In one way or another, these issues are all associated with the users' perception of the system.

2.6.3 Dialogue Strategies in AdApt

AdApt uses a mixed initiative dialogue strategy. The division of initiative between agent and user varies between tasks. The initial task is to find out if there are any apartments in the database that suit the user's requirements. The interaction in this task is mainly system led. If the number of objects (apartments) that match the user's constraints exceeds seven, the system tries to reduce the number of hits through a set of constraining questions. When the number of hits is 7 or less the apartments are displayed on a map as coloured dots. The system's constraining questions were implemented to study referring expressions in AdApt (Skantze, 2000). The apartments displayed as coloured dots on the map were used to facilitate the use of reference expressions. The number seven was chosen since it is considered to be the amount of elements that one person can hold in his/her working memory (Ashcraft, 1994) The agent's constraining questions include requests to make the user add values such as size, area, price and apartment attributes.

- | | |
|-------|--|
| 2.R-1 | Jag vill ha en lägenhet på söder. I want an apartment in Södermalm. |
| 2.S-1 | Hur många rum vill du ha? How many rooms do you want? |
| 2.R-2 | Två rum. Two rooms. |
| 2.S-2 | Hur mycket får lägenheten kosta? How much can the apartment cost? |
| 2.R-3 | En komma åtta miljoner. 1.8 million Skr |
| 2.S-3 | Är det något speciellt du vill att din lägenhet ska ha? Is there something special that you want in your apartment? |
| 2.R-4 | Balkong. Balcony. |
| S-4 | Det finns 7 sådana lägenheter och de visas nu på kartan. There are seven such apartments and they are displayed on the map. |

As soon as the number of hits is 7 or less the apartments are displayed on the map. The user has the opportunity to choose to not specify any constraints by replying "no" or "it does not matter" to the constraining questions. If the system runs out of questions and the number of hits still exceeds 7, 7 randomly chosen apartments, which fulfil the constraints that were given, will be displayed on the map. This is most likely not an ultimate strategy since the number of hits actually exceeds 7 and the system out-put will be misleading. However, this strategy was chosen to not force the users to add constraints that they did not want to specify. The system led dialogue strategy has weaknesses and is possibly not the most efficient one. For example: if the user choose not to specify the values that the system asks for, the dialogue costs will increase in terms of elapsed time, user turns and system turns. However, another strategy could be used in future AdApt tests.

When a number of apartments are already in focus, the dialogue is completely user led. The user is free to ask questions about the apartments and no initiative is taken by the system:

- 3.R-1 Hur stor är lägenheten på Bellmansgatan tjuogoåtta?
How large is the apartment on Bellmansgatan 28?
- 3.S-1 Den gröna lägenheten har 1 rum
The green apartment has 1 room.
- 3.R-2 Hur stor är den blåa lägenheten
How large is the blue apartment?
- 3.S-2 Den blå lägenheten har 2 rum
The blue apartment has 2 rooms.
- 3.R-3 Hur mycket kostar den blå lägenheten
How much is the blue apartment?
- 3.S-3 Den blå lägenheten kostar 1440000 kronor
The blue apartment costs 1440000 Skr.
- 3.R-4 Hur många kvadratmeter är den blå lägenheten
How many square metres is the blue apartment?
- 3.S-4 Den blå lägenheten är på 65.3 kvadrat
The blue apartment is 65.3 square metres.

2.6.4 Turn-taking gestures and hourglasses in a multi-modal dialogue system

Two different systems set-ups were created to study how well “busy gestures” compared to a symbol perform in preventing users from speaking when the system is preparing a reply. Furthermore, which one of the two system versions is more efficient and which technique appeals more to the users? Peoples take turns speaking in human-human dialogues. The changing of turns in human-human dialogue is normally a smooth process which rarely breaks down due to simultaneous speech or interruptions (Torres, Cassell & Prevost, 1997). Turn-taking most likely involves some non-verbal exchange between users. Information to facilitate turn-taking can be provided by hand gestures, body posture, speech, gaze or different combinations of these. Clearly, the use of such information would be of great use in human-machine dialogue. AdApt sometimes take a little while to generate a response and the user may well decide to say something during this time. This is a problem since AdApt does not support barge-ins. Barge-ins are difficult for the speech recognizer to deal with and even if this was possible the best course of action may well be to generate the response even if the user barges in. In cases like this, the dialogues would run smoother if the users wait for their turn before speaking. There are various strategies to use for doing this. The strategies tested were: one system version with a symbol suggesting that the system is busy (e.g. an hourglass) and one in which the agent uses facial gestures to show that he is busy thinking. For a more detailed description see Edlund & Nordstrand (2002).

2.7 Using PARADISE on AdApt

Wizard of Oz methodology has earlier been used in the development of AdApt. The limitations of this methodology that has been discussed earlier have led to a need for an evolutionary tool which is capable of evaluating the system version available. Since AdApt is now a fully working system with functional components a simulation of functionalities is no longer needed and a new method for evaluation is desirable. The research topics in AdApt are related to the users' perception of the system and PARADISE is a general framework for evaluation with the overall goal to maximize user satisfaction. Consequently, PARADISE appears to be suitable for evaluating AdApt's performance. Furthermore, its abilities to specify the relative contribution of different factors to user satisfaction, compare different dialogue strategies and to compare different system versions needs to be further investigated to determine how successful PARADISE is in doing this. As a first step we applied PARADISE to AdApt.

2.7.1 Using open tasks

The contribution of evaluation is to pinpoint weaknesses of dialogue strategies used in the system's natural context. To obtain a useful corpus, the interaction with the system has to include some sort of task. To procure reliable data the tasks should mirror scenarios that could be similar to real life events. However, it is questionable whether predetermined scenarios are representative of real scenarios (Larsen, 1999). In order to obtain a dialogue corpus that reflects the behaviour of real users we used open tasks. The tasks were open in that they were defined by the users. This was done in order to represent real tasks better than predetermined scenarios would do. The users were instructed to: "find information about apartments that you might want to buy, live in or be otherwise interested in". To facilitate representations of task success and the employment of open tasks, we chose not to represent our corpus in a confusion matrix. However, data from the AdApt corpus could be represented in confusion matrixes but this would be time consuming and result in inconveniently large confusion matrices.

The apartment market in Stockholm is of current interest and concerns most of its population. For this reason we believe that the users were capable of defining their own tasks. However, the use of open tasks left us with the tricky task to define tasks and determine task success from the system logs. According to our task definition, a new task was always initiated by the user and ended with either a system response or a change of subject by the user. Our definition of tasks and task success are described in more detail in section 3.

PROMISE, a Procedure for Multi-modal Interactive System Evaluation, is a method for evaluating multi-modal spoken dialogue systems developed by Beringer, Kartal, Louka, Schiel and Türk in the SmartKom project. The methodology is partly based on PARADISE, but introduces a number of novelties. The new aspect of PROMISE is an attempt to create a general framework for evaluation of multi-modal dialogue systems and to address the problems of defining tasks and task success for systems. As opposed to PARADISE, PROMISE uses a rather imprecise task definition in order to make the interaction between user and system as natural as possible (Beringer, Louka, Penide-Lopez & Türk, 2002). The task scenarios were therefore not defined in advance and not represented in attribute value matrixes. In order to normalise for task complexity PROMISE extract a number of superordinate concepts depending on the task at hand. The superordinate concepts are termed "information bits". According to the authors the: "...information bits are carefully selected, categorized and weighted by hand before the test start". The details of these categorizations are not mentioned in the paper. However, the definitions of tasks and markings of task success seem to be similar to the methods used in the evaluation of AdApt.

Multi-modality

AdApt is a multi-modal system and PARADISE had to be adjusted to include this specific feature. Multi-modality was considered through adding five questions to the questionnaire. Two questions were related to the system's overall graphical output and three questions were related to the different system contributions. The adjustment to multi-modality also included adding an extra dialogue cost metric: the number of mouse clicks per utterance. The mouse clicks and the system's graphical output were also used for definition of tasks and task success.

Choice of metrics

The set of cost metrics used in the evaluation of AdApt was chosen in order to suit the specific features of the system. For example: a number of metrics that could not be used because of the characteristics of the system's architecture were excluded. Another set was added. A more detailed description of these cost metrics used in this study is provided in section 3.

Finally, there are probably more ways PARADISE could have been adjusted to suit the specific characteristics of AdApt, both in terms of the choice of metrics and definition of tasks and task success. However, we tried to stay close to the PARADISE methodology in order to obtain an initial corpus and weights for the performance function since we had no earlier experience of using the method. The corpus collected can later be used as experimental data to further investigate how PARADISE can be modified to suit AdApt.

2.7.2 Purpose

The overall goal of AdApt is to study human-computer interaction in a multi-modal conversational dialogue system. The main purpose of this study was to investigate whether PARADISE was suitable as an evaluation tool for such systems. The data collected for this purpose was made in collaboration with another separate study: "Turn-taking gestures and hourglasses in a multi-modal dialogue system" (Edlund & Nordstrand, 2002). PARADISE was also applied to make comparisons between the different system contributions in this test.

3 Method

The data collection was made from dialogues with 26 subjects. The subjects were all between 20 and 40 years of age. Half of the subjects were women. The groups were balanced for gender, but otherwise randomly distributed. None of the subjects had any professional experience of speech technology, although the majority claimed to have used some sort of speech interface at some point. Furthermore, the majority of the users stated that they had some experience with computers and had used an apartment search tool on the web. All subjects claimed a reasonable knowledge of the geography of downtown Stockholm, which is the area that the AdApt system is concerned with. None of the users had used or seen AdApt before the experiment. Each subject received two cinema tickets as appreciation of their participation.

3.1 Pre-tests

Two pre-tests were made. The first pre-test was made in order to test whether it was possible to use open tasks and the second test was made to test the instructions and the questionnaires.

Pre-test 1

The first pre-test was made to make sure that the task was not too difficult. If the subjects can not obtain any information from the system the tests will be of no use. Pre-test one included two subjects recruited from the Royal Institute of Technology in Stockholm. Both subjects were master's thesis students at the Department of speech, music and hearing, but neither of them had any experience of using AdApt. The instructions were deliberately sparse in order to not lead the subjects in their interaction with the system. The results from pre-test 1 showed that that the task was difficult but not impossible. The subjects came across difficulties, but they managed to obtain information from the system and despite occasional problems, they were capable to continue the interaction with the system. The decision to use open tasks was therefore sustained.

Pre-test 2

The purpose of the second pre-test was to test if the instructions and the questionnaires were easy to understand, and to make sure that they were interpreted as they were intended. The test included three subjects who had no professional experience of speech technology. The instructions did not result in any problems and no changes were made. An interview after the test showed that one of the questions regarding the pace of the system was equivocal and was therefore excluded.

3.2 Experimental design

Each subject carried out a dialogue using one of three available system versions. A between-subject design was used with 26 subjects split into 3 groups. Each group interacted with one of the three system set-ups: facial turn-taking gestures, an hourglass symbol and a configuration with no visual turn-taking feedback at all. Configuration type was an independent variable in this study.

The metrics used in the experiment were collected through three different methods: (1) user surveys, (2) recordings of the dialogues and (3) automatic logging of the system's dialogue behaviour.

3.3 Equipment

3.3.1 AdApt

The AdApt system consists of graphical interface including a map and an animated agent (the agent is given the name Urban). The user can use both speech and mouse clicks as input. The system uses speech as output. Furthermore, the system output includes graphical markings that indicate where on the map the apartments are situated and highlights the apartments in focus.



Figure 3.1: AdApt

3.3.2 DAT-recorder

A DAT-recorder was used to record the entire interaction between the subjects and the system. The recordings were used as a complement to the system's recording. This was done in order to capture the utterances that were not registered by the system.

3.3.3 Video camera

A video camera was used as a supplementary visual recording device. Since the subjects were alone during the interaction the video recordings were used as an extra device in the study. However, the videotapes were never used in this analysis of the data.

3.4 Data Collection

The tests were performed in the premises of the Department for speech, music and hearing at the Royal Institute of Technology in Stockholm. Before the test the subjects received the following information about the system:

- The system takes mouse and voice input
- The system output is voice synthesis from an animated talking head and dots on a map.
- It is quite possible to get stuck in some situations. If this happens, say “Urban, börja om” (“Urban, start over”), Urban is the name given to the animated agent used in the system.
- The system has information about apartments for sale in downtown Stockholm.
- The task is to find information about apartments that you might want buy or live in.

The subjects were also informed that with this little guidance, at least the first minutes of the test would be difficult. The simple drawing of the experimental setup is provided in appendix 8.2. The test begun when the examiner left the room. To help the subjects get a smooth start and to set the scene for the rest of the dialogue the subjects were told to initiate the interaction with the utterance: “Hej Urban” (Hello Urban!). This utterance triggered Urban to shortly introduce himself and the system functionalities (see appendix 8.5). Each subject talked to the system for a little over 30 minutes. After that time they were interrupted by the examiner. The last task which was being carried out when they were interrupted was excluded from the data analysis. The subjects were aware that they were recorded with an open microphone and a video camera. No further assistance was given during the test (except for someone entering the room silently to restart a crashed module on a couple of occasions), and after the test, conversation with them was kept to a minimum until they had filled out the questionnaires.

3.4.1 *The metrics*

The dependent variables used in the experiment are based on PARADISE and are a combination of dialogue quality measures, dialogue efficiency measures and task success (SUCCESS). However, some changes were made to adjust the metrics to the specific features of AdApt. The changes were made to adjust the framework to AdApt’s different modalities, its specific dialogue structure and its open tasks. The questionnaires were used as a measure of **user satisfaction** (US).

Dialogue Costs

The **dialogue efficiency** metrics contributes to the dialogue costs and were calculated from the sound recordings and the system logs. The dialogue efficiency metrics used in the experiments were:

| | |
|----------|--|
| #TASKS | Tasks per subject. |
| AVUTT | Total number of utterances/#Tasks |
| %USERUTT | Total number of user utterances / Total number of utterances (system and user) |
| AVWORD | Total number of user words /Total number of user utterances |
| AVCLICKS | Total number of user clicks (mouse) /#Tasks |
| ELTIME | Elapsed time/# Tasks |

The other contributors to dialogue costs are the **dialogue quality** metrics. The dialogue quality metrics are derived from the hand-labelled dialogues, the system logs and the sound recordings:

| | |
|--------------|---|
| %UTTUNHEARD | Total number of unheard user utterances/Total number of user utterances |
| %WORDUNHEARD | Total number of unheard user words /Total number of user words |
| %WER | Word Error Rate/Total number of words registered by the system |
| %ACC | Word Accuracy/Total number of words registered by the system |
| AVCQUERY | Total system constraint queries /# Tasks |
| AVREPROMPT | Total system reprompts (for explanation see 2.2.2 Disambiguates p.13) /#Tasks |
| %CANCEL | Total number of user cancels/#Tasks |
| AVDIAFAIL | Total dialogue failure responses /#Tasks |
| AVAREA | Total number of area markings with the mouse /#Tasks |
| AVUREP | Total number of user repetitions /#Tasks |
| AVUCORR | Total number of user corrections /#Tasks |
| AVUHELP | Total number of request for help/#Tasks |
| AVUDISRUPTED | Total number of disrupted user utterances /#Tasks |
| AVUASKREPEAT | Total number of user requests for repetitions/ #Tasks |
| AVUDISFLUENT | Total number of disfluent utterances /#Tasks |

Task success was also used as a dependent variable in the multivariate linear regression analysis:

| | |
|----------|---|
| SUCCESS | #Successful tasks/#Tasks (1 point of task success per successful task) |
| WSUCCESS | #Successful tasks/#Tasks (1 point of task success given for each unit of information exchanged in successful tasks) |

Word error rate (%WER) is a frequently used metric for measuring the performance of the speech recognizer component. There are three types of recognition errors: substitutions, deletions and insertions. Substitutions are replaced words. Deletions are words left out and insertions are inserted words that do not correspond to the words that were actually spoken by the user. Word error rate is defined as the sum of all three types of errors divided by the total number of words in the reference transcript.

Other quality metrics that are typically used in PARADISE are barge-ins, help requests and timeout prompts. However, since this version of AdApt does not support barge-ins and does not have any timeout prompts or help functionality these metrics were not used. Barge-ins and help requests could have been hand labelled and used as a quality metric, but the transcriptions of the dialogues showed that the user rarely made requests for help and rarely barged in over the system. Consequently, these two cost metrics were never used in the final multiple linear regression analysis. Furthermore, many of the barge-ins that actually occurred were probably not even intended to be “heard” by the system. They were often out of the domain utterances (meta utterances) and spoke in a silent manner. Oppermann, Schiel, Steininger and Beringer (2001) have termed this kind of meta utterances *Off-Talk*.

User Satisfaction

The user **satisfaction measures** were collected through a user survey (appendix 8.4). The survey included the following questions:

- Förstod du vad Urban sa? **TTS Performance**
(Was Urban easy to understand?)
- Förstod Urban vad du sa? **ASR Performance**
(Did Urban understand what you said?)
- Var det enkelt att hitta information om lägenheterna? **Task Ease**
(Was it easy to find information about the apartments?)
- Visste du vad du skulle säga till Urban? **User Expertise**
(Did you know what to say to Urban?)
- Var systemet för långsamt och slött med att svara? **System Response**
(Was the system too sluggish and slow to reply to you?)
- Betedde sig Urban som du hade förväntat dig? **Expected Behaviour**
(Did Urban behave as you expected him to?)

- Utifrån din nuvarande erfarenhet av systemet (AdApt) tror du att du skulle använda det för att hitta bostadsrätter? **Future Use**
(From you current experiences with using the system (AdApt) do you think you would use the system to find apartments?)

The main part of the questions used in the survey is a translation of the questions used in PARADISE. Some of the questions had to be slightly adjusted to suit the domain of AdApt. The following changes were made:

(1) The usability measure **Comparable Interface** was excluded since this study was not a comparison between systems.

(2) The user survey was complemented with two extra questions to capture the multi-modal features of AdApt. The two extra questions were:

- Hade du nytta av det grafiska gränssnittet för att få den information du ville ha?
(Was the graphical interface useful to obtain the information you wanted?)
- Tyckte du om det grafiska gränssnittet?
(Did you like the graphical interface?)

(3) Four more questions were added to obtain user satisfaction ratings that were related to the different system contributions:

- Tyckte du att interaktionen med systemet fungerade bra?
(Did you think that your interaction with the system worked smoothly?)
- Var det lätt att avgöra om systemet väntade på att du skulle prata?
(Was it easy to determine if the system was waiting for you to say something?)
- Var det lätt att avgöra om systemet var upptaget med att formulera ett svar?
(Was it easy to determine if the system was busy generating a respond?)
- Tyckte du att systemet tog lång tid på sig att formulera ett svar?
(Did you think that the system was slow to respond?)

All question responses, except two, ranged from the value “nej, nästan aldrig” (no, almost never) to “ja, nästan alltid” (yes, almost always). Each response was mapped to an integer. In the end of the questionnaire there was room for free comments. In PARADISE a questionnaire is completed after each task. However, this could not be done with user defined tasks. We were restricted to use the questionnaire only once, after the complete test. The user satisfaction score in this survey is therefore an overall evaluation of the dialogue and cannot be associated with specific tasks. This is a weakness in our definition of user satisfaction. However, it was not possible to collect more detailed user satisfaction scores since we used open tasks. A user satisfaction score was calculated for the dialogue through summing the scores from the multiple-choice questions in the survey.

3.4.2 Hand Labelling

PARADISE relies mainly on measures that can be logged automatically. However, some of the metrics require hand labelling. The metrics that were hand labelled in this test were: cancels (%CANCEL), reprompts (AVREPROMPT), requests for repetitions (user), system repetitions and user repetitions (AVUREP). The main part of the transcriptions and definitions of the tasks were also made by hand. The first set of transcriptions was made using a graphical transcription tool. The transcription tool displayed the string of words as interpreted by the speech recogniser. The recognized string of words was compared to the sound recordings made by the system and the errors were corrected by hand. The transcription tool was used at the same time to label meta utterances, cut-offs and disfluencies. However, the transcriptions based on the system recordings were not complete since AdApt does not support barge-ins and is not always in “listening mode”. Occasionally the subjects spoke when the system was not “listening”. For example the subjects sometimes spoke when the system was about to generate a response and these utterances were not registered by the system. These “unheard” utterances were captured by the DAT-recorder. The first cycle of transcriptions was complemented with a second cycle based on these recordings.

The transcribed user utterances were also hand labelled with tags that were used for calculating the qualitative metrics. The labelled tags were reprompts, cancels, user repetitions, system repetitions, meta utterances, and requests for repetitions. The **reprompts** (AVREPROMPT) used in AdApt are:

- (1) “Jag förstår inte vad du menar kan du formulera om?” (Sorry, I do not understand what you mean. Please rephrase!)
- (2) “Jag förstår inte vilken lägenhet du menar” (I do not understand which apartment you mean.).

All user utterances that expressed a request for a restart of the system were classified as **cancels** (%CANCEL). User utterances with the same semantic content as the previous user utterance were marked as **user repetition** (AVUREP). **Meta utterances** (AVUMETA) were user utterances that were out of domain. Finally, **requests for repetition** (AVUASKREPEAT) were labelled. Requests for repetition were when the user asked the system to repeat an utterance.

Task definition

The subjects were instructed to search for information about apartments that they might want to live in, buy or had other interests in. The transcriptions of the dialogues were used to classify tasks. Meta utterances, cancels and requests for repetitions were defined as out-of-domain utterances. Out-of-domain utterances were excluded from the task definition. Task definition was made according to the following rules: (1) a new task is always initiated by the user. (2) A task is ended either through a system response (correct or incorrect) or change of subject made by the user. A task is also ended when the user provides the system with some kind of contradicting information. This will be discussed in more detail shortly. A new task is initiated when the user makes a request for new information (within the system domain):

- | | |
|-------|---|
| 4.R-1 | Hur många kvadratmeter är den skära lägenheten? How many square meters has the pink apartment? |
| 4.S-1 | Den skära lägenheten är på 47 kvadrat. The pink apartment has 47 square metres. |

The subject in this example initiates a new task and the system responds. The response is correct and the task is considered to be successful. To determine task success the system's answers are compared to values stored in the database. If the answer is wrong, i.e. the pink apartment turns out to have 58 square metres instead of 47; the system utterance: "The pink apartment has 47 square metres" can still be considered to be the end of the task. Whether it does or not depends on the next user utterance. For example: If the user does not recognise that the answer is wrong and makes a request for new information, i.e. change of subject, the system response in the dialogue example is still considered as task-ending. However, if the user repeats the question, i.e. he or she might have recognised the answer as being wrong; the system response will not be considered to be the end of the task. This strategy was also used for correct answers.

- 5.R-1 Hur mycket kostar den vita lägenheten?
How much is the white apartment?
- 5.S-1 Den gula lägenheten ligger på våning 5.
The yellow apartment is on floor 5.
- 5.R-2 Hur mycket kostar den vita lägenheten?
How much is the white apartment?
- 5.S-2 Den vita lägenheten kostar 850000 kronor.
The white apartment costs 850000 SEK.

The first system response (5.S-1) is wrong and is not considered to end the task, since the user repeats the same question in the next utterance (5.R-2). However, the next system utterance, 5.S-2, is the correct answer and is also considered to be the end of the task.

User utterances that contradict the information provided earlier in the same task are also considered to be the end of a task. However, this might not always be the intention of the user, but the strategy was used to keep the definition of tasks consistent. We needed to be consequent since we did not want to speculate about the users' intentions.

- 6.R-1 Jag vill ha en lägenhet.
I want an apartment.
- 6.S- 1 Hur många rum vill du ha?
How many rooms do you want?
- 6.R-2 Fyra rum.
Four rooms.
- 6.S-2 Hur mycket får lägenheten kosta?
How much do you want to pay for the apartment?
- 6.R-3 Tre miljoner.
Three million Skr.
- 6.S-3 Var ska lägenheten ligga?
Where do you want your apartment?
- 6.U-1 På kungsholmen.
In Kungsholmen.
- U-2 På södermalm.
On södermalm.

This is an example of when the user provides the system with contradicting information. Note that all the user responses in the dialogue, except for the last one, only add constraints. However, the last user utterance, U-2 (U=unregistered by the system), contradicts U-1 since the apartment can possibly be situated both in Kungsholmen and Södermalm. U-2 is therefore considered as the first utterance of a new task. The user might want to keep constraints given earlier in the dialogue (for example the price and the size). Whether the user intends to keep the constraints given earlier or not is impossible to know. U-2 is therefore classified as the initiation of a new task with area, Södermalm, as the only constraint. This approach was used to keep the task definition consistent.

The 26 dialogues resulted in 1737 tasks. The tasks were subsequently tagged as being correct or incorrect. The main objective in PARADISE is to maximize user satisfaction, the correctness or incorrectness of the task was solely based on the system output, since the output is all the information that the user gets acquainted with. If the user utterance was correctly “understood” by the system, i.e. how the utterance was interpreted by the speech recogniser, the parser or the dialogue manager, was consequently not taken in consideration when defining task success. AdApt sometimes interpret a user utterance wrongly but for some reason still manage to respond correctly. An incorrect system interpretation of the user’s utterance is consequently not considered and the task is classified as successful only depending on the system’s output only. Consequently, the system might very well generate the right answer the “wrong” way and still receive points for task success.

Task Success

AdApt carries out a set of different tasks in which different kinds of information are exchanged. The problem is how to make comparisons between tasks with different degrees of task complexity. PARADISE typically uses predefined tasks and comparisons between tasks are not a problem since the Kappa coefficient normalizes for task complexity. However, in this study the tasks were defined by the users and each subject performed a unique set of tasks. Tasks could no longer be easily compared. If no consideration is taken to what kind of information that is exchanged, tasks where many “units” of information are exchanged will in the end most likely be treated as less successful since they will receive higher dialogue costs. For example:

- 7.R-1 Har du några lediga lägenheter i Gamla Stan?
Do you have any apartments in Gamla Stan?)

- 7.S-1 Det finns 3 sådana lägenheter och de visas nu på kartan
There are 3 such apartments that are displayed on the map

This is a successful task where the system returns the correct value. However, the next dialogue is also a successful task:

- 8.R-1 Jag skulle vilja ha en lägenhet på södermalm.
I would like to have an apartment on Södermalm.
- 8.S-1 Hur många rum vill du ha?
How many rooms do you want?
- 8.R-2 Två.
Two.
- 8.S-2 Hur mycket får lägenheten kosta?
How much do you want to pay for the apartment?
- 8.R-3 Tre miljoner.
Three millions.
- 8.S-3 Är det något speciellt du vill att din lägenhet ska ha?
Is there something special that you want in your apartment?
- 8.R-4 Badkar.
A bathtub.
- 8.S-4 Det finns 7 sådana lägenheter och de visas nu på kartan.
There are 7 such apartments that are displayed on the map.

Both tasks are successful, but the second one consists of 8 utterances where the first one only consists of 2. The second one will receive higher dialogue costs in terms of more utterances and more words. However, the extra utterances are not worthless since valuable information is exchanged. As an attempt to compensate for the increased dialogue costs in tasks where more units of information are exchanged are given a higher task success rating. Each task receives 1 point per “unit” of information exchanged. The first task example consequently received 2 points (apartment and Gamla Stan) and the second task received 5 points (apartment, Södermalm, two rooms, three million, and bath tub). However, if the user chose not to specify a value which the system made a request for no extra points will be given. These utterances will only be counted as dialogue costs since no valuable information is exchanged. For example:

- 9.R-1 Visa mig alla lägenheter med tre rum.
I want to see all apartments with three rooms.
- 9.S-1 Hur mycket får lägenheten kosta?
How much do you want to pay for the apartment?
- 9.R-2 Spelar ingen roll.
It does not matter.
- 9.S-2 Var ska lägenheten ligga?
In what area do you want your apartment?
- 9.R-3 På kungsholmen.
In Kungsholmen.
- 9.S-3 Det finns 7 sådana lägenheter och de visas nu på kartan.
There are 7 such apartments that are displayed on the map

This (successful) task received only 3 points. For some reason, the user does not want to specify the price and no units of information are exchanged in utterances S-1 and R-2 and they receive no points of task success. The user also has the possibility to specify several values in one utterance. For example:

10.R-1 Jag skulle vilja ha en lägenhet på Södermalm med tre rum och en balkong
I would like an apartment in Södermalm with three rooms and a balcony

If system answers correctly the task will receive 4 points of task success and very low dialogue costs. This is reasonable since this is an efficient strategy if it is successful and the successful exchange of information should be rewarded. However, if tasks like these always fail the compensation for task success will have no positive impact in the results.

Graphical markings

The graphical output was also taken in consideration in the definition of task success. AdApt's graphical output displays where on the map the apartments that match the users' constraints are situated. The apartments are marked as coloured dots and the name of the colours can be used to refer to the apartments (i.e. "Has the blue apartment got a balcony?"). Each apartment in the database has a unique id-number. When an apartment is displayed on the map, its id-number is automatically registered in the log files. The id-numbers in the log files were later used to check if the apartments that were displayed on the map matched the users' constraints. The mouse could also be used to mark areas on the map and the coordinates of these markings were also automatically logged. The coordinates of these markings were later used to verify that the apartments that were displayed on the map were situated inside the coordinates.

4 Results

The 26 dialogues were classified in 1737 different tasks. Furthermore, the dialogues consisted of 6431 spoken utterances of which 2043 (12749 words) were system utterances and 4388 were user utterances. The total number of spoken words were 26586 (system=12749 and user=13837).

The dialogue costs (efficiency metrics + quality metrics) and the user satisfaction ratings were normalized to their Z scores. This was done to ensure that the magnitude of the coefficients in the regression equation would reflect the relative contribution of that factor.

Multiple linear regression presumes that the predictor variables are independent of each other. A correlation matrix was made to ensure that the variables in the test were not too strongly correlated. The test revealed a number of correlations between the metrics and the following variables were excluded: AVUREP, AVUMETA, %WORDUNHEARD, %ACC and AVRESTART. System restarts (AVRESTART) correlated with user cancels (AVCANCEL), word accuracy (%ACC) correlated with word error rate (%WER), the number of unheard words (%WORDUNHEARD) correlated with unheard utterances (%UTTUNHEARD), the number of meta utterances (AVUMETA) correlated with (%UTTUNHEARD) and the number of user repetitions (AVUREP) correlated with the number of utterances per task (AVUTT). To avoid correlations between the number of utterances per task, the number of constraint queries, the number of reprompts and dialogue failure responses were calculated per utterance instead of per task as first was intended. Furthermore, elapsed time was not used as efficiency metric since we did not have time to calculate for this variable. However, elapsed time per task would have been strongly correlated with the number of utterances per task (AVUTT) and it would most likely be problematic to use both of these metrics.

After these dependencies had been removed, multiple regressions were performed with User Satisfaction as the dependent variable and the remaining cost metrics and task success as the independent variables. The first set of multiple linear regressions was made on four different sets of data. This was done to study the results of using two different sets of questions in the questionnaire and two different markings of task success. The first test was applied to user satisfaction for questions 1-8 and 1 point of task success for each correct task:

| | Coefficient | P-Value |
|---------|--------------------|----------------|
| AVUTT | -0.041 | 0.0025 |
| AVQUERY | 0.040 | 0.0035 |
| SUCCESS | 0.530 | <0.0001 |

Table 1

The statistically significant (0.05 levels of significance) predictors of User Satisfaction in this test were (AVUTT), (AVQUERY) and (SUCCESS).

| | Coefficient | P-Value |
|-----------|--------------------|----------------|
| AVUTT | -0.029 | 0.0205 |
| %USERUTT | -0.034 | 0.0151 |
| AVDIAFAIL | -0.023 | 0.0288 |
| SUCCESS | 0.036 | 0.0034 |

Table 2

The set of data tested were User Satisfaction for questions 1-8 and 10-13 and 1 point of task success given for each correct task. The statistically significant (0.05 levels of significance) predictors of User Satisfaction in this test were (AVUTT), (%USERUTT), (AVDIAFAIL) and (SUCCESS).

| | Coefficient | P-Value |
|----------|--------------------|----------------|
| AVUTT | -0.043 | 0.0017 |
| AVQUERY | 0.033 | 0.0144 |
| WSUCCESS | 0.041 | 0.0014 |

Table 3

The set of data tested were User Satisfaction for questions 1-8 and 1 point of task success given for each unit of information exchanged in correct tasks. The statistically significant (0.05 levels of significance) predictors of User Satisfaction in this test were (AVUTT), (AVQUERY) and (WSUCCESS).

| | Coefficient | P-Value |
|-----------|--------------------|----------------|
| AVUTT | -0.030 | 0.0162 |
| %USERUTT | -0.038 | 0.0043 |
| AVDIAFAIL | -0.026 | 0.0155 |
| WSUCCESS | 0.029 | 0.0125 |

Table 4

The set of data tested were User Satisfaction for questions 1-8 and 10-13 and 1 point of task success given for each unit of information exchanged in correct tasks. The statistically significant (0.05 levels of significance) predictors of User Satisfaction in this test were (AVUTT), (%USERUTT), (%AVDIAFAIL), and (WSUCCESS).

A repeated multivariate linear regression was made on the statistically significant predictors of User Satisfaction for each set of data:

| | Coefficient | P-Value |
|-----------|--------------------|----------------|
| AVUTT | -0.373 | <0.0001 |
| %USERUTT | -0.158 | <0.0001 |
| AVQUERY | -0.195 | <0.0001 |
| AVDIAFAIL | -0.145 | <0.0001 |
| SUCCESS | -0.640 | 0.589 |

1). Question 1-8, 1 point of task success per successful task

| | Coefficient | P-Value |
|-----------|--------------------|----------------|
| AVUTT | -0.363 | <0.0001 |
| %USERUTT | -0.141 | <0.0001 |
| AVQUERY | -0.170 | <0.0001 |
| AVDIAFAIL | -0.127 | <0.0001 |
| SUCCESS | -0.103 | 0.0026 |

2). Question 1-8, 10-13, 1 point of task success per successful task

| | Coefficient | P-Value |
|-----------|--------------------|----------------|
| AVUTT | -0.321 | <0.0001 |
| %USERUTT | -0.138 | <0.0001 |
| AVQUERY | -0.174 | <0.0001 |
| AVDIAFAIL | -0.006 | 0.8471 |
| WSUCCESS | -0.047 | 0.1835 |

3). Question 1-8, 1 point of task success given for each unit of information exchanged in successful tasks

| | Coefficient | P-Value |
|-----------|--------------------|----------------|
| AVUTT | -0.311 | <0.0001 |
| %USERUTT | -0.121 | 0.0003 |
| AVQUERY | -0.148 | <0.0001 |
| AVDIAFAIL | 0.012 | 0.7161 |
| WSUCCESS | -0.085 | 0.0151 |

4). Question 1-8, 10-13, 1 point of task success given for each unit of information exchanged in successful tasks.

A stepwise regression was made to make confirm the results. The results made no difference. A multivariate linear regression was also made on the different system contributions. The results for the different groups had several differences (see appendix 8.6). In the group with gestures there were two significant predictors of User satisfaction across all four sets of data: SUCCESS (task success) and AVDIAFAIL. %USERUTT were significant in all sets except for the set with questions 1-8 and 1 point of task success per information unit exchanged in successful tasks. In the group with the hour-glass symbol the only significant (positive) predictor of User satisfaction for all four sets of data was SUCCESS. %USERUTT was a significant predictor four all sets of data except for the one with questions 1-8 and 1 point of task success per successful task. The significant predictors of user satisfaction in the group with no gestures were AVUTT and AVQUERY across all four sets of data.

5. Discussion

This study was an attempt to assess PARADISE's capacity as an evolution tool for AdApt. We were interested to see if the method was capable of capturing multi-modality and the use of open tasks. These goals will now be discussed on the basis of the experiences we made during the tests and in the lights of the results.

5.1 The performance of PARADISE

Initially a few general things can be said about PARADISE. The purpose of the method is to take three different aspects of evaluation, task success, user satisfaction and dialogue costs, in consideration and bring them together in an overall performance function. The authors of PARADISE declare the importance and the need for a general evaluation framework and specify how the different contributors to user satisfaction, dialogue costs and task success, can be normalized to enable comparisons. Furthermore, the method provides a procedure for how task success can be determined, a set of cost metrics and a questionnaire to collect user satisfaction scores. However, the central role of task definition in PARADISE and the importance of constructing a throughout questionnaire are not discussed in detail. The task definition issue will be considered here shortly.

Multivariate linear regression is used to combine the dialogue costs and to specify how multiple factors contribute to User Satisfaction. The purpose of using multivariate linear regression is to make an overall performance evaluation without being forced to make hypothesis about the outcome in advance. However, multivariate linear regression presupposes that the different variables are independent of each other. This is problematic since a number of dialogue costs in PARADISE are correlated, some even per definition. In an evaluation of three spoken dialogue systems for e-mail access, ANNIE, ELVIS and TOOT, Walker, Kamm & Litman (2000) use the total number of tasks (#task), the absolute number of cancels (#cancel) and the percentage of cancels (%cancel) per task. These metrics are clearly related and should not be used as predictors of the same independent variable in multivariate linear regression.

The first set of multivariate linear regressions in our study was made without considering the possible occurrence of correlating variables. The results from these tests were not robust; they varied across tests and metrics. To test if correlating variables were interfering and as an attempt to overcome the problem we tried to use metrics that were as independent as possible of each other. A correlation matrix was made and variables that were highly correlated with other variables were excluded. A new set of multiple linear regressions was performed and, with a few minor exceptions, all four sets of data had the same predictive variables of User satisfaction. This suggests that multivariable linear regression can be used when there are no strong correlations between the predictor variables. However, there are most likely other, less obvious, dependencies between the metrics used in PARADISE. For example: It seems plausible that the quality of the dialogue and the efficiency of the dialogue are somehow related to each other. The possible occurrence of depending variables can restrict the designer in the choice of metrics. He/she will have to start to make hypothesis about the outcome of the evaluation in terms of what metrics are related. Unfortunately this was just what PARADISE was trying to avoid. We suggest that a correlation matrix is used to reveal correlations between metrics for each data collection.

Our results from the multivariate linear regression say something about the performance of PARADISE. If the metrics that were shown to be significant predictors of user satisfaction are plausible factors of usability there are reasons to believe that the evaluation was successful. The results from the multivariate linear regression showed that there were three predictors that were statistically significant for all four sets of data. %USERUTT and AVUTT were negative predictors of User Satisfaction and SUCCESS was a positive predictor. Their effects in the multivariate linear regressions seem reasonable. For example: If the percentage of user utterances per total number of utterances (%USERUTT) is high, the percentage of system utterances is relatively low. A low percentage of system utterances indicate that turn-taking between user and system is not working satisfactory (the user receives few responses). A high percentage of user utterances should consequently have a negative effect on User satisfaction and this is also what the multivariate linear regressions suggest. The efficiency metric AVUTT (#utterances /#tasks) was also a negative predictor of User Satisfaction. Tasks with a large number of utterances are consequently perceived as inefficient and tiresome. According to earlier results from applying PARADISE efficiency has not been a significant predictor of User satisfaction. According to the authors of PARADISE their results: "...draw into question the frequently made assumption that efficiency is one of the most important measures of system performance" (Walker, Litman, Kamm & Abella, p.26, 1998). Despite results from earlier studies, the number of user utterances seems to affect User satisfaction in this particular study. However, it seems reasonable that the relationship is not linear. A dialogue with no utterances is not optimal and neither is a dialogue with a very large number of utterances. However, the use of linear regression presumes that the relationship between the criterion variable and the predictor variable is linear (Hinkle, Wiersma & Jurs, 1998). AVUTT's influence on User Satisfaction will be discussed more thoroughly in relation to the different system contributions. SUCCESS (task success) positive effect on User satisfaction is also predictable. If the users carry out successful tasks, their perception of the system will most likely be positive. Since the effects of the metrics in the test were reasonable, it lends support to the usefulness of PARADISE.

5.1.2 Definition of tasks and task success

The definition of tasks and task success is essential in PARADISE. The purpose of evaluating a SDS should therefore be to evaluate the system in terms of user satisfaction related to the designer's goals of the system. The overall goal of PARADISE is to maximize user satisfaction, but satisfied users are of no use if they do not perform actions that the system was designed to help them complete. The system goals are reflected in the tasks. The tasks should therefore mirror actions that real users will carry out and the successful performance of more valuable tasks should receive higher points of task success. Consequently, if we believe that the strategy of using constraining questions leads to exchange of valuable information in AdApt such successful tasks should be rewarded. It is the successful exchanges of attributes that are rewarded, not the strategy for doing this. Which marking strategy should be used depends on the goals of the system. If the goal of AdApt was to sell apartments, the constraining questions can be fruitful if they manage to help the user specify constraints which results in a small number of hits, apartments. A small number of apartments can easily be compared and the comparison can help the user to make further discriminations between the apartments to finally choose one apartment which he/she wants to buy. However, the purpose of AdApt is not to sell apartments and the extra units of information that are exchanged in the constraining part of the dialogue should probably not be given extra points.

The results from the different marking strategies of task success are difficult to interpret. AVQUERY (#constraining questions/#utterances) is a positive predictor of User satisfaction for three of the data sets: (1) question 1-8 and 1 point of task success for each successful task, (2) question 1-8, 10-13 and 1 point of task success for each successful task and (3) question 1-8 and 1 point of task success for each information unit in successful tasks. This suggests that the extra points of task success given in two of the data sets did not have any noteworthy effect on User satisfaction. However, multivariate linear regressions for the different system contributions (see appendix 8.6) showed that AVQUERY only had a positive effect on User Satisfaction in the group with no gestures. This effect was statistically significant across all four sets of data in the no-gestures group, and only of this group. These results possibly suggest that when no gestures that support turn-taking are available, the system led part of the dialogue works more smoothly compared to the part of the dialogue which is user led. This seems fairly reasonable since the user receives more straight forward feedback in the constraining phase than in the rest of the dialogue when no busy gestures are available. The constraining phase is consistent and the user knows what kind of behaviour to expect from the system. However, during the user-led phase of the dialogue, the status of the system can be perceived as less obvious.

PARADISE has mainly been applied to goal-oriented systems. It is relatively easy to define tasks and reference answers in advance for these systems, since they are designed to perform comparatively well-structured actions. However, definition of tasks and task success in less goal-oriented systems is more complex and using predefined tasks may restrict the user in her/his interaction with the system. For example: if the subjects in the AdApt data collection were asked to look for a certain apartment it is possible that their only focus would be to find this particular apartment. They would not look around to see what was available as a normal user might do. One way to overcome the problem is to let the acceptable values in the cells of the attribute value matrix to be more than one. However, if many values are accepted in each cell in the AVM the confusion matrix will very soon grow to be inconveniently large. Still, it is questionable whether corpus collected using this strategy is representative of real users' dialogue behaviour.

The task definition issue is not solely related to PARADISE, it is an issue that has to be considered in all kinds of dialogue corpus collections. We tried to let the users define their own tasks in order to obtain dialogues that were representative of dialogues from real users. In Walker, Hirschman & Aberdeen (2000) PARADISE was applied to 7 fixed and 2 open task scenarios. The mixture of fixed and open tasks was used to see if there were any significant differences between data collected using user-defined tasks from data collected using pre-defined tasks. The results showed that the open tasks were somehow problematic. The users were more inclined to change their destination halfway through, in the open task scenarios than in the fixed task scenarios.

This tendency was also present in this study. However, it is difficult to know what caused the changes of attributes. This dialogue is an example of a user who changes between attributes in AdApt:

- 11.U-87 Det är jättebra. Har den balkong?
That is very good. Has it got a balcony?
- 11.R-87 Har den balkong?
Has it got a balcony?
- 11.U-88 Har den sjöutsikt?
Has it got a view of the lake?
- 11.S-56 Ja, lägenheten har balkong.
Yes the apartment has got a balcony
- 11.U-89 Den gula, har den gula lägenheten balkong?
The yellow one, has the yellow apartment got a balcony?
- 11.R-88 Har den gula lägenheten balkong?
Has the yellow apartment got a balcony?
- 11.U-90 Har lägenheten sjöutsikt?
Has it got a view of the lake?
- 11.S-57 Nej, den gula lägenheten har inte balkong.
No, the yellow apartment has not got a balcony.

This particular dialogue is very unfortunate. The bad performance is primarily due to bad speech recognition. The user utterances marked with a U (unheard) are not registered by the system. The subject keeps changing back and forth between the attributes. It is very likely that the subject recognizes the system's response S-56 as wrong, since the system does not use the name of the colour (yellow) to verbally refer to the apartment. This behaviour conveys two problems. First: Are open task scenarios really representative of real task scenarios as was intended? I.e., does the subjects' dialogue behaviour mirror the behaviour of real users? Secondly: Is our definition of task success appropriate for this kind of dialogue behaviour? The dialogue receives no points of task success despite the fact that the subject actually obtains the information he/she was asking for at the end of the dialogue ("No, the yellow apartment has not got a balcony").

When searching for an apartment you might look for one or two vivid attributes, while other attributes are not as important. What attributes that are important most certainly vary across users and it is therefore difficult to make generalizations. The change of attributes does not necessarily mean that the subjects have given up on finding a suitable apartment. Furthermore, the task in this study did not have a particular goal. The users were told to look around to see what was available and that is possibly what they were doing. We also noted that many of the subjects who changed attributes tended to go back and ask for the same attribute later in the dialogue. The dialogue example earlier is an illustration of this. This kind of behaviour supports the view that the users were still eager to find out the value of a particular attribute. If their only interest had been to generate a response they would not keep repeating questions about the same attributes over and over again.

Definition of tasks was based on already existing dialogue corpus. This was difficult, since it is impossible to know the intention behind a user utterance. Based on the structure of the dialogues the transcriptions were classified into tasks. The definition of tasks was made according to a set of predefined rules, which have been described earlier. Sticking to these rules, the interpretations of some utterances were still ambiguous, and a few utterances were most certainly defined incorrectly. However, to be as consistent as possible the tasks were "defined" and marked by two judges. The

agreement between the two judges was almost absolute. More than 1500 tasks were defined and less than 30 differed across the judges. Some tasks were presumably defined in favour of the system, but just as many seemed to receive fewer points than they deserved. To study the consequences of different task definitions and points of task success in more detail different definitions and markings should be tested in order to see how different interpretations affect the outcome. This was done to some extent but the area needs to be further investigated to set up more throughout rules for task definition. However, the inner intentions of the subjects will still be inaccessible and definitions of tasks and task success will never be totally objective. The decision to use open tasks scenarios is depending on whether one believes that the advantages of user-defined tasks exceed the disadvantages.

In a more throughout evaluation of the graphical input time should have been included. The duration in time between a user utterance (or a system request) and a mouse click is important for the system's semantic interpretation. The timing also effects task definition. The definition should be depending on whether a click is made before or after an utterance. Moreover, the time duration between a mouse-click and an utterance should be crucial for whether they should be interpreted in combination or independently. However, the dialogue management in AdApt is fairly complex and it is difficult to time the different components of the system. Therefore no consideration to time was taken except for how utterances and tasks were related chronologically. This can have affected task success, since a few tasks were most likely incorrectly judged. However, the number of graphical inputs in the data collection was relatively small and the affect was probably too small to influence the system's overall performance.

It is reasonable to believe that the system's dialogue behaviour has a strong effect on the user's perception of the system. Still, the cost metrics in PARADISE say little about different dialogue behaviours. Walker & Passonneau (2001) suggest a Dialogue-Act Tagging Scheme, DATE, to be used in combination with PARADISE. The purpose of DATE is to provide a finer-grained quantitative dialogue metric for evaluating DARPA COMMUNICATOR, a platform for spoken dialogue system. A similar scheme for AdApt would be useful to make evaluation more profitable. Unfortunately, there was no time in this study to develop such a scheme. To create a dialogue act-tagging scheme for AdApt would be an interesting future area of research.

5.1.3 Multi-modality

Since multi-modality is a central research issue in the AdApt project an evaluation method for the system has to take this particular feature in consideration. However, the only metric that was not related to speech, AVCLICKS (mouse clicks), was not a significant predictor of user satisfaction. This does not necessarily suggest that PARADISE is not capable of capturing the relative contributions of different modalities in AdApt. It is possible that the mouse clicks did not effect the users' perception of the system either way. The mouse was relatively rarely used compared to speech. Furthermore, using the mouse requires only a small effort and when it is not working this might only have a little or no negative effect on User Satisfaction.

The different system contributions are related to the system's visual output. The multivariate linear regressions for the different system versions showed several differences. First, SUCCESS (#task success/#tasks) was a statistically significant predictor of User satisfaction for all four sets of data in the gesture- and the hour-glass group, but for none of the data sets in the group with no gestures. SUCCESS which is a strong positive predictor of US in the other groups is far from significant in the group with no gestures. This can possibly be interpreted as that not even the performance of

successful tasks manages to please the users when there are no busy-gestures available. This is interesting since the group with no gestures was expected to get frustrated because of the minimal amount of feed-back. Another interesting tendency in the results from the multivariate linear regression for the different system contributions is that the only predictor of US across all sets of data in the group with the hour-glass is SUCCESS. The hour-glass was expected to be efficient in terms of illustrating the status of the system. However, the users might perceive the system as slow and the hour-glass as unnatural. Consequently, the only thing that “makes them happy” is when they manage to successfully carry out tasks.

The group who interacted with the system version with gestures had two positive predictors of US (User Satisfaction) across all four sets of data: AVDIAFAIL and (W)SUCCESS. (W)SUCCESS had a very strong positive effect on US compared to the other metrics and the other groups. The positive effect of AVDIAFAIL on User satisfaction is difficult to interpret. AVDIAFAIL is a system failure which results in the system response “Jag har inget svar just nu” (I have not got an answer right now). This is consequently an indication of that something has gone wrong, which intuitively should be perceived as something negative. However, the users often seem to be unconvinced that their utterances have been registered by the speech recognition component and are possibly frustrated when this is not confirmed. The utterance “I have not got an answer right now” may be seen as a confirmation of the utterance being registered, possibly even correctly registered, and therefore perceived as something positive, despite the fact that they do not receive the information that they were asking for. To reveal more reliable results of the differences between the different system contributions it is necessary to use the overall performance function. This will be done but since the main purpose of this study was not to compare the different system contributions these results will not be presented here.

Despite the fact that the purpose of this study was not to evaluate AdApt’s overall performance, it would be reasonable to say a few things about it. Unfortunately, except for calculating for the different system contributions, which have already been done, there is not much to say about the system’s overall performance based on this single study. For example SUCCESS (task success) and AVUTT (#utterances/#task), which are a positive respectively a negative predictor of User satisfaction. It is clear that one would like to increase task success and reduce the number of utterances per task. However, the cause of SUCCESS’ and AVUTT’s effects can be difficult to interpret based on this single study. It is easier to reveal the cause of the metric’s effect if a comparison between two system versions is made. For example: two different dialogue strategies are compared and User satisfaction is higher for one of the strategies. According to PARADISE it then would be reasonable to say that the choice of dialogue strategy have an effect on User satisfaction, given that it is the only difference between the two system versions.

6 Future research

Evaluation plays a crucial role in all kinds of spoken dialogue system development. PARADISE have challenged the problems of revealing the relative contributions of different metrics and combined task success, user satisfaction and dialogue costs in an overall performance function. However, these methods still require further experimentation. Issues to be considered are definitions of tasks that are representative of “real” tasks, marking of task success and evaluation of multi-modal systems. The authors of PARADISE suggest that the framework can be used on multi-modal systems. The systems in their tests, however, are uni-modal, speech-only systems. Further research is needed to test whether PARADISE is capable of mirroring the relative contributions of different modalities. PROMISE (SmartKom) suggests a number of characteristics in multi-modal systems that are not considered in PARADISE. These features include weighting the different components in multi-modal spoken dialogue systems in terms of their purpose and recognition accuracy. Furthermore, PROMISE uses unspecified tasks and a method for normalisation of task success in terms of “information bits”, which is similar to the one used in the evaluation of AdApt. Walker and Litman (2000) make an attempt to use open tasks in PARADISE. The definition of open tasks and normalisation of task complexity is consequently an issue of current interest. It is essential to create a general framework for evaluation of spoken dialogue systems that is independent of a system’s domain, choice of modalities, complexity and purpose. Task definition and usability should be given central roles in a framework for evaluation.

An interesting start in addressing these general research issues would be to further explore the data already collected in this study. The data can be used to test how different definitions of tasks and different markings of task success affect the outcome. Another interesting continuation of this study would be to make a second evaluation with a different system version to compare the results from that study with the results from this study. To be able to do this it would be necessary to state throughout rules for definition of open tasks and a set of core metrics. This can preferably be done based on the data and results from this study. As mentioned earlier it would also be interesting to develop a dialogue act tagging scheme for AdApt to create finer-grained quantitative dialogue metric. Preferably such a scheme would be able to define the relative contributions of different modalities. Finally, there is still much to do in terms of efficiency. Several of the metrics which were now hand labelled could be computerized and a throughout log file standard would make evaluation easier and less time consuming.

7. Acknowledgements

Special thanks go to Jens Edlund for providing me with so much help. Jens has been a great support and assistance all through the work. For being involved with the user tests of AdApt I would like to thank Jens Edlund and Magnus Nordstrand. I would also like to thank the users who managed to engage in a “meaningful” conversation with Urban. Finally I would like to thank my examiner Rolf Carlson and Arne Jönsson, associate professor at Linköping University, for helping me to gain an overall perspective of my work.

8 References

- Antoine, J-Y., Siroux, J., Caelen, J., Villaneau, J., Goulian, J., Ahafhaf, M., (2000) Obtaining Predictive Results with an Objective Evaluation of Spoken Dialogue Systems: Experiments with the DCR assessment Paradigm, *Proceedings of the Second International Conference on Language Resources and Evaluation*, Athens,
- Benoît, C. (1992). The intrinsic bimodality of speech communication and the synthesis of talking faces. *Journal on Communications*, 43, Scientific Society for Telecommunications, Hungary (July-Sept), 32- 40
- Bellik, Y., (1994) Modality Integration: Speech and Gesture, *LIMSI-CNRS*, Orsay, France
<http://cslu.cse.ogi.edu/HLTsurvey/ch9node6.html#SECTION94> (2002-04-15)
- Bernsen, N. O., Dybkjær, L., (2000) Is that a good spoken language dialogue system? In Jacquemin, C., Mariani, J. and Paroubek, P. (Eds.): In *Proceedings of the Workshop Using Evaluation within HLT Programs: Results and Trends*. Conference on Language Resources and Evaluation (LREC'2000), Athens, Greece http://www.class-tech.org/publications/5_SLDS-eval-15.5.00-F.doc (2002-05-13)
- Beskow, J., (1995) Rule-based Visual Speech Synthesis, In *Proceedings of Eurospeech '95*, Madrid, Spain
- Boyce, S. J, Gorin, A. L., (1996) User interface issues for natural spoken dialogue systems, *Proceedings of ISSD 96 (International Symposium on Spoken Dialogue)*, pp.65-68.
- Bickmore, T. and Cassell, J. (2001) Relational Agents: A Model and Implementation of Building User Trust, In *Proceedings of CHI-2001*, pp. 396--403, ACM Press, New York
- Cassell, J., Bickmore, T., Billinghurst, M., Campbell, L., Chang, K., Vilhjálmsson, H., Yan, H. (1999) Embodiment in Conversational Interfaces: Rea, CHI99, Pittsburgh, PA.
<http://citeseer.nj.nec.com/cassell99embodiment.html>
- Cassell, J. (2000) More than just another pretty face: Embodied conversational interface agents.
<http://citeseer.nj.nec.com/cassell00more.html> (2002-08-15)
- Danieli, M., Gerbino, E., (1995), Metrics for evaluating dialogue strategies in a spoken language system. In *Proceedings of the 1995 AAAI Spring Symposium on empirical methods in discourse interpretation and generation*
- Giachin, E., (1995), Spoken Language Dialogue, *CSELT*, Torino, Italy
<http://cslu.cse.ogi.edu/HLTsurvey/ch6node6.html> (2002-04-15)
- Gustafson, J., Bell, L., Beskow, J., Boye, J., Carlson, R., Edlund, J., Granström, B., House, D., Wirén, M., (2000) AdApt – A Multi-modal Conversational Dialogue System In an Apartment Domain, In *Proceedings of ICSLP 00*

Lamel, L., Rosset, S., and Gauvain, J. L., (2000) Considerations in the Design and Evaluation of Spoken Language Dialogue Systems, In *Proc. ICSLP'2000*, pages IV-5-8, Beijing, <http://citeseer.nj.nec.com/423911.html> (2002-05-10)

Litman, D., Pan, S., and Walker, M., A., (1998) Evaluating Response Strategies in a Web-based Spoken Dialogue Agent. In *Proc. ACL/COLING 98: 36th Annual Meeting of the Association of Computational Linguistics*, pages 780--787

Litman, D., Pan, S, (1999) Empirically Evaluating an Adaptable Spoken Dialogue System
AT&T Labs - Research, Florham Park, NJ, USA
Computer Science Department, Columbia University, New York, NY, USA
<http://www.cs.usask.ca/VM99/Proc/litman.pdf>, (2002-07-02)

McGee, D. R., Cohen, P.R., and Oviatt, S., (1998) Confirmation in multi-modal systems, In *Proceedings of the International Joint Conference of the Association for Computational Linguistics and the International Committee on Computational Linguistics (COLING- ACL '98)*, Montreal, Quebec, Canada

McTear, M., F., (2002) Spoken Dialogue Technology: Enabling the Conversational User Interface, pp.1-85 *ACM Computing Surveys (CSUR)*, ACM Press New York, NY, USA
(ISSN:0360-0300)

Oppermann, D., Schiel, F., Steininger, S. and Beringer, N., (2001) Off-Talk – a Problem for Human-Machine-Interaction? In *Proceedings of EUROSPEECH 2001*, Scandinavia, Aalborg.

Oviatt, S., De Angeli, A., Kuhn, K., (1997), Integration and synchronization of input modes during multi-modal human-computer interaction. In *Proceedings of the workshop: Referring Phenomena in a Multimedia Context and their Computational Treatment*, ACL/EACL'97, July 11, 1997, Madrid. 1-13.
<http://citeseer.nj.nec.com/oviatt97integration.html>

Oviatt, S. L., (2000) Multi-modal Interface Research: A Science Without Borders, In B. Yuan Huang & X. Tang (Eds.), In *Proceedings of the International Conference on Spoken Language Processing (ICSLP'2000)*, Vol. 3, (pp. 1-6). Beijing

Paek, T., (2001) Empirical Methods for Evaluating Dialogue Systems, *SIGdial*
http://www.sigdial.org/sigdialworkshop01/papers/paek_long.pdf (2002-03-08)

Skantze, G., (2000), Koordinering av refererande uttryck i multi-modal människa-datordialog. *MA thesis, LIU-KOGVET-D-0036-SE*, Linköping University, Sweden

Skantze, G. (2002). Coordination of referring expressions in multimodal human-computer dialogue. In *Proceedings of ICSLP 2002*

Sturm, S., Wang F., and Cranen, B., (2001) Adding Extra Input/Output Modalities to a Spoken Dialogue System, In *Proceedings 2nd ACL SIGdial Workshop on Discourse and Dialogue*, Aalborg, Denmark

Torres, O., Cassell, J., and Prevost, S., (1997). Modelling Gaze Behaviour as a function of discourse structure. *First international workshop Human-Computer conversation*. Bellagio, Italy

Turunen, M., Hakulinen, J., (2001) Agent-based Error Handling in Spoken Dialogue Systems
In *Proceedings of the Eurospeech 2001*: 2189-2192

Walker, M. A., Litman, D. J., Kamm, C. A. and Abella, A., (1997) PARADISE: A Framework for Evaluating Spoken Dialogue Agents. In: *Proc. of the 35th Annual Meeting of the Assoc. for Computational Linguistics (ACL/EACL 97)*, Morgan Kaufmann, USA-San Francisco, 271-280.

Walker, M. A., Litman, D. J., Kamm, C. A. and Abella, A., (1998) Evaluating Spoken Dialogue Agents with PARADISE: Two case studies, *Computer Speech and Language*, vol. 12-3

Walker, M. A., Kamm, C. A., & Litman, D. J. (2000). Towards developing general models of usability with PARADISE, *Natural Language Engineering: Special Issue on Best Practice in Spoken Dialogue Systems*.
<http://citeseer.nj.nec.com/article/walker00towards.html>

Walker, M. A., Kamm, C., Boland, J., (2000) Developing And Testing General Models Of Spoken Dialogue System Performance *Language Resources and Evaluation Conference (LREC)*,
<http://citeseer.nj.nec.com/296479.html>

Walker, M., Hirschman, L., and Aberdeen, J., (2000) Evaluation for DARPA Communicator Spoken Dialogue Systems, *Language Resources and Evaluation Conference, LREC*,
<http://fofoca.mitre.org/sites/MITRE/papers/walker00evaluation.pdf> (2002-02-21)

Walker, M., Passonneau, R., (2001) DATE: A Dialogue Act Tagging Scheme for Evaluation of Spoken Dialogue Systems, In *Proceedings of Human Language Technology Conference*, San Diego,
<http://www.research.att.com/~walker/dtag6.pdf> (2002-02-12)

Books

Ashcraft, M., H., (1994) *Human Memory and Cognition*
New York: HarperCollins. Prentice Hall
(ISBN 0673467899)

Clark, C., (1997) *Using Language*
Cambridge University Press, Cambridge
(ISBN 0 521 56745 9)

Hinkle, D., E., Wiersma, W. and Jurs, S., G., (1998), *Applied Statistics for the Behavioural Sciences*,
Houghton Mifflin Company, Boston
(ISBN 0-395-87411-4)

9 Appendix

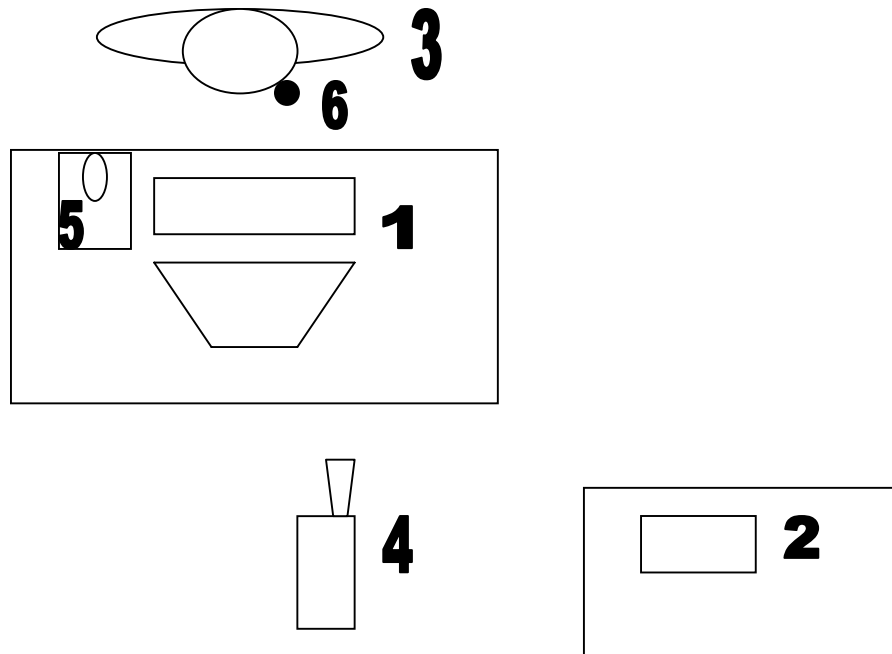
9.1 Confusion Matrix

(Walker, Litman, Kamm and Abella, 1998)

| | KEY | | | | | | | | | | | | | | | |
|------|-------------|----|----|----|--------------|----|----|----|--------------|--|--|-----|-------------|-----|-----|-----|
| | Depart City | | | | Arrival City | | | | Depart Range | | | | Depart Time | | | |
| DATA | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | | | V10 | V11 | V12 | V13 | V14 |
| V1 | 16 | | 1 | | 4 | | | | 3 | | | 2 | | | | |
| V2 | 1 | 20 | 1 | | | 3 | | | | | | | | | | |
| V3 | 5 | 1 | 9 | 4 | 2 | | 4 | | | | | | | | | |
| V4 | 1 | 2 | 6 | 6 | | | 2 | | | | | | | | | |
| V5 | 4 | | | | 15 | | | | 2 | | | 3 | | | | |
| V6 | 1 | 6 | | | | 19 | | | | | | | | | | |
| V7 | | | 5 | 2 | 1 | 1 | 15 | 4 | | | | | | | | |
| V8 | | 1 | 3 | 3 | 1 | 2 | 9 | 11 | | | | | | | | |
| V9 | 2 | | | | 2 | | | | 39 | | | 10 | | | | |
| V10 | | | | | | | | | 6 | | | 35 | | | | |
| V11 | | | | | | | | | | | | | 20 | 5 | 5 | 4 |
| V12 | | | | | | | | | | | | | | 10 | 5 | 5 |
| V13 | | | | | | | | | | | | | 5 | 5 | 10 | 5 |
| V14 | | | | | | | | | | | | | | 5 | 5 | 11 |
| SUM | 30 | 30 | 25 | 15 | 25 | 25 | 30 | 20 | 50 | | | 50 | 25 | 25 | 25 | 25 |

9.2 Experimental set-up

1. Screen
2. DAT-recorder
3. Subject
4. Video camera
5. Mouse
6. Microphone



9.3 Dialogue cost metrics

Efficiency metrics

| | |
|----------|---|
| #TASKS | Tasks per subject. |
| AVUTT | Total number of utterances/#Tasks |
| %USERUTT | Total number of system utterances/ Total number of utterances (system and user) |
| AVWORD | Total number of user words /Total number of user utterances |
| ELTIME | Elapsed time/# Tasks |
| AVCLICKS | Total number of user clicks /#Tasks |

Quality metrics

| | |
|--------------|---|
| AVCQUERY | Total system constraint queries /# Tasks |
| %UTTUNHEARD | Total number of unheard user utterances/Total number of user utterances |
| %WORDUNHEARD | Total number of unheard user words /Total number of user words |
| %WER | Word Error Rate/Total number of heard words |
| %ACC | Word Accuracy / Total number of heard words |
| AVREPROMPT | Total system reprompts/#Tasks |
| %CANCEL | Total number of user cancels/#Tasks |
| AVDIAFAIL | Total dialogue failure responses /#Tasks |
| AVAREA | Total number of area markings /#Tasks |
| AVUREP | Total number of user repetitions /#Tasks |
| AVUCORR | Total number of user corrections /#Tasks |
| AVUMETA | Total number of meta utterances /#Tasks |
| AVUDISRUPTED | Total number of disrupted user utterances /#Tasks |
| AVUDISFLUENT | Total number of disfluent utterances /#Tasks |
| AVUASKREPEAT | Total number of user requests for repetitions/ #Tasks |
| AVUFRAGMENT | Total fragment utterances /#Tasks |

9.4 User Survey

1. Förstod du var Urban sa?

Nej, nästan aldrig Sällan Ibland Ja, oftast Ja, nästan alltid

2. Förstod Urban vad du sa?

Nej, nästan aldrig Sällan Ibland Ja, oftast Ja, nästan alltid

3. Var det enkelt att hitta information om lägenheterna?

Nej, mycket svårt Nej, svårt Sådär Ja, lätt Ja, mycket lätt

4. Visste du vad du skulle säga till Urban?

Nej, nästan aldrig Sällan Ibland Ja, oftast Ja, nästan alltid

5. Var systemet för långsamt och slött med att svara?

Nej, nästan aldrig Sällan Ibland Ja, oftast Ja, nästan alltid

6. Betedde sig Urban som du hade förväntat dig?

Nej, nästan aldrig Sällan Ibland Ja, oftast Ja, nästan alltid

7. Utifrån din nuvarande erfarenhet av systemet (AdApt) tror du att du skulle använda det för att hitta bostadsrätter?

Absolut inte Troligen inte Kanske Troligen Absolut

8. Hade du nytta av det grafiska gränssnittet för att få den information du ville ha?

Nej, nästan aldrig Sällan Ibland Ja, oftast Ja, nästan alltid

9. Tyckte du om systemets grafiska gränssnitt?

Ja Nej

10. Tyckte du att interaktionen med systemet fungerade bra?

Nej, nästan aldrig Sällan Ibland Ja, oftast Ja, nästan alltid

11. Var det lätt att avgöra om systemet väntade på att du skulle prata?

Nej, nästan aldrig Sällan Ibland Ja, oftast Ja, nästan alltid

12. Var det lätt att avgöra om systemet var upptaget med att formulera ett svar?

Nej, nästan aldrig Sällan Ibland Ja, oftast Ja, nästan alltid

13. Tyckte du att systemet tog lång tid på sig att formulera ett svar?

Nej, nästan aldrig Sällan Ibland Ja, oftast Ja, nästan alltid

9.5 System introduction

Hej jag heter Urban! Jag har information om bostadsrätter som är till salu i Stockholms innerstad. Du kan ställa frågor och peka på kartan så får vi se vad jag hittar sedan kan vi prata mer om lägenheterna. Om du vill börja om från början säger du bara: "Urban börja om!" så gör jag det.

Hello, my name is Urban! I have information about apartments that are for sale in Stockholm. You can ask me questions and click on the map and we will see what I find. Then we can talk some more about the apartments. If you want to start over just say: "Urban, start over!" and I will do that.

9.5 Multivariate Linear Regressions for the three system contributions

9.5.1 Gestures

| | Coefficient | P-Value |
|-----------|--------------------|----------------|
| AVUTT | -0.013 | 0.5209 |
| %USERUTT | 0.042 | 0.0184 |
| AVQUERY | 0.038 | 0.0377 |
| AVDIAFAIL | 0.040 | 0.0082 |
| SUCCESS | 0.106 | <0.0001 |

1). Question 1-8, 1 point of task success per correct task

| | Coefficient | P-Value |
|-----------|--------------------|----------------|
| AVUTT | -0.018 | 0.3831 |
| %USERUTT | 0.029 | 0.1008 |
| AVQUERY | 0.026 | 0.1657 |
| AVDIAFAIL | 0.035 | 0.0252 |
| SUCCESS | 0.081 | <0.0001 |

2). Question 1-8, 1 point of task success per correct task

| | Coefficient | P-Value |
|-----------|--------------------|----------------|
| AVUTT | 0.003 | 0.8701 |
| %USERUTT | 0.035 | 0.0143 |
| AVQUERY | -0.012 | 0.4168 |
| AVDIAFAIL | 0.025 | 0.0458 |
| SUCCESS | 0.079 | <0.0001 |

3). Question 1-8, 10-13, 1 point of task success given for each unit of information exchanged in correct tasks

| | Coefficient | P-Value |
|-----------|--------------------|----------------|
| AVUTT | -0.001 | 0.9746 |
| %USERUTT | 0.028 | 0.0470 |
| AVQUERY | -0.022 | 0.1408 |
| AVDIAFAIL | 0.022 | 0.0772 |
| SUCCESS | 0.068 | <0.0001 |

4). Question 1-8, 10-13, 1 point of task success given for each unit of information exchanged in correct tasks.

9.5.2 Hour-glass

| | Coefficient | P-Value |
|-----------|--------------------|----------------|
| AVUTT | -0.028 | 0.1102 |
| %USERUTT | -0.028 | 0.1009 |
| AVQUERY | 0.007 | 0.6802 |
| AVDIAFAIL | -0.006 | 0.6808 |
| SUCCESS | 0.052 | 0.0023 |

1). Question 1-8, 1 point of task success per correct task

| | Coefficient | P-Value |
|-----------|--------------------|----------------|
| AVUTT | -0.031 | 0.0788 |
| %USERUTT | -0.036 | 0.0301 |
| AVQUERY | 0.002 | 0.9293 |
| AVDIAFAIL | -0.009 | 0.5134 |
| SUCCESS | 0.043 | 0.0079 |

2). Question 1-8, 1 point of task success per correct task

| | Coefficient | P-Value |
|-----------|--------------------|----------------|
| AVUTT | -0.032 | 0.0749 |
| %USERUTT | -0.039 | 0.0271 |
| AVQUERY | 0.019 | 0.3080 |
| AVDIAFAIL | -0.022 | 0.1282 |
| SUCCESS | 0.049 | 0.0053 |

3). Question 1-8, 10-13, 1 point of task success given for each unit of information exchanged in correct tasks

| | Coefficient | P-Value |
|-----------|--------------------|----------------|
| AVUTT | -0.035 | 0.0530 |
| %USERUTT | -0.045 | 0.0078 |
| AVQUERY | 0.013 | 0.4704 |
| AVDIAFAIL | -0.024 | 0.0856 |
| SUCCESS | 0.043 | 0.0100 |

4). Question 1-8, 10-13, 1 point of task success given for each unit of information exchanged in correct tasks.

9.5.3 No gestures

| | Coefficient | P-Value |
|-----------|--------------------|----------------|
| AVUTT | -0.070 | <0.0001 |
| %USERUTT | -0.019 | 0.3526 |
| AVQUERY | 0.058 | 0.0032 |
| AVDIAFAIL | -0.005 | 0.8539 |
| SUCCESS | -0.006 | 0.7472 |

1). Question 1-8, 1 point of task success per correct task

| | Coefficient | P-Value |
|-----------|--------------------|----------------|
| AVUTT | -0.070 | <0.0001 |
| %USERUTT | -0,019 | 0.3638 |
| AVQUERY | 0.058 | 0.0027 |
| AVDIAFAIL | -0.004 | 0.8663 |
| SUCCESS | -0.006 | 0.7918 |

2). Question 1-8, 10-13, 1 point of task success per correct task

| | Coefficient | P-Value |
|-----------|--------------------|----------------|
| AVUTT | -0.049 | 0.0046 |
| %USERUTT | 0.003 | 0.8810 |
| AVQUERY | 0.058 | 0.0034 |
| AVDIAFAIL | -0.007 | 0.7813 |
| WSUCCESS | -0.009 | 0.6559 |

3).Question 1-8, 1 point of task success given for each unit of information exchanged in correct tasks

| | Coefficient | P-Value |
|-----------|--------------------|----------------|
| AVUTT | -0.049 | 0.0047 |
| %USERUTT | 0.003 | 0.8675 |
| AVQUERY | 0.058 | 0.0028 |
| AVDIAFAIL | -0.007 | 0.7863 |
| WSUCCESS | -0.010 | 0.6483 |

4).Question 1-8, 10-13, 1 point of task success given for each unit of information exchanged in correct tasks.