Modeling Modality Selection in Multimodal Human-Computer Interaction

Extending Automated Usability Evaluation Tools for Multimodal Input

vorgelegt von
M.A.
Stefan Schaffer
geb. In Kassel

von der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
– Dr.-Ing. –

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender:	Prof. Dr.-Ing Reinhardt Karnapke
Gutachter:	Prof. Dr.-Ing. Sebastian Möller
Gutachter:	Prof. Dr. Manfred Thüring
Gutachterin:	Prof. Dr. Kristiina Jokinen

Tag der wissenschaftlichen Aussprache: 03. Juni 2016

Berlin 2016

Stefan Schaffer

Modeling Modality Selection in Multimodal Human-Computer Interaction

Extending Automated Usability Evaluation Tools for Multimodal Input

June 10, 2016

For Sonja, Lino and Pepe.

Abstract

In this work the following three basic research questions are discussed: (1) can significant effects of modality efficiency and input performance on the selection of input modalities in multimodal HCI be disclosed by unified experimental investigations? (2) Can a utility-driven computational model of modality selection be formed based on empirical data? (3) Can the compiled model for modality selection be utilized for the practical application in the field of automated usability evaluation?

Initially, foundations of decision-making in multimodal HCI are discussed, and the state of the art in automatic usability evaluation (AUE) is described. It is shown that there are currently no uniform empirical results on factors influencing modality choice that allow for the creation of a computational model. As part of this work two AUE tools, the MeMo workbench and CogTool, are extended by a newly created computational model for the simulation of multimodal HCI.

Aiming at answering the first research question, the empirical part of the thesis describes three experiments with a mobile application integrating touch screen and speech input. In summary the results indicate that modality efficiency and input performance are important moderators of modality choice.

The second research question is answered by the derivation of a utility-driven model for input modality choice in multimodal HCI based on the empirical data. The model provides probability estimations of modality usage, based on different levels of the parameters modality efficiency and input performance. Four variants of the model that differ in training data are tested. The analysis reveals a considerable fit for models based on averaged modality usage data.

Answering the third research question it is illustrated how the modality choice model can be deployed within AUE tools for simulating multimodal interaction. The multimodal extension as well as the practical utilization of MeMo is depicted, and it is described how unimodal CogTool models of touch screen and speech based interaction can be rendered into multimodal models. A comparison of data generated by simulations with the AUE tools with predictions of the derived modality selection algorithm verifies the correct integration of the model into the tools. The practical application discloses the usefulness of the modality choice model for the prediction of the number of steps and the total time spent to solve specific tasks with

multimodal systems. The practical part is concluded by a comparison of Memo and CogTool. Both tools are classified, and an assessment on a subjective basis as well as on the the basis of the quality of predictions is conducted.

Summary and outlook condense the added value provided by the thesis, and identify starting points for future work.

Zusammenfassung

In dieser Arbeit werden die folgenden drei grundlegenden Forschungsfragen diskutiert: (1) können durch vereinheitlichte experimentelle Untersuchungen signifikante Auswirkungen der Modalitäteneffizienz und der Performanz der Eingabemodalität auf die Auswahl von Eingabemodalitäten in multimodaler Mensch-Computer Interaktion offen gelegt werden? (2) kann auf der Grundlage empirischer Daten ein Utility-gesteuertes Rechenmodell der Modalitätenwahl gebildet werden? (3) Kann das erstellte Modell für die Modalitätenwahl auf dem Gebiet der automatischen Bewertung der Benutzerfreundlichkeit (Usability) praktisch eingesetzt werden?

Zunächst werden Grundlagen der Entscheidungsfindung in multimodaler Mensch-Computer-Interaktion diskutiert, und der Stand der Technik in der automatischen Usability Evaluation (AUE) beschrieben. Es wird aufgezeigt, dass bisher keine einheitlichen empirischen Ergebnisse zu Einflussfaktoren der Modalitätenwahl vorhanden sind, die die Erzeugung eines Rechenmodells ermöglichen. Im Rahmen dieser Arbeit werden zwei AUE Werkzeuge, die MeMo Werkbank und CogTool, durch ein neu erstelltes Computermodell zur Simulation von multimodaler Mensch-Computer-Interaktion erweitert.

Mit dem Ziel, der Beantwortung der ersten Forschungsfrage, werden im empirischen Teil der Arbeit drei Experimente mit einer mobilen Anwendung beschrieben, welche Touchscreen und Spracheingabe integriert. Zusammenfassend zeigen die Ergebnisse, dass sowohl die Modalitäteneffizienz als auch die Performanz der Eingabemodalitäten wichtige Einflussfaktoren der Modalitätenwahl sind.

Die zweite Forschungsfrage wird durch die Herleitung eines Utility-gesteuerten Modells zur Auswahl von Eingabemodalitäten in multimodaler Mensch-Computer-Interaktion auf der Grundlage der empirischen Daten beantwortet. Das Modell liefert Wahrscheinlichkeitsschätzungen der Modalitätennutzung, basierend auf unterschiedlichen Niveaus der Parameter Modalitäteneffizienz und Performanz der Eingabemodalität. Vier Varianten des Modells, die sich in den Trainingsdaten unterscheiden, werden untersucht. Die Analyse zeigt eine beträchtliche Vorhersagegüte für die Modelle die auf der gemittelten Modalitätennutzungsdaten basieren.

Zur Beantwortung der dritten Forschungsfrage wird dargestellt, wie das Modell für die Modalitätenwahl zur Simulation multimodaler Interaktion in AUE Werkzeu-

gen eingesetzt werden kann. Die Erweiterungen für Multimodalität sowie die praktische Nutzung von MeMo werden aufgezeigt, und es wird beschrieben, wie unimodale CogTool Modelle für Touchscreen- und sprachbasierte Interaktion in multimodale Modelle überführt werden können. Ein Vergleich der Daten, die durch Simulation mit den AUE Werkzeugen erzeugt werden, mit den Vorhersagen des hergeleiteten Modalitätwahlalgorithmus, bestätigt die korrekte Integration des Modells in die Werkzeuge. Bei der praktischen Anwendung durch die AUE Werkzeuge erweist sich die Nutzbarkeit des Modells für die Vorhersage der Anzahl von Schritten und der Bearbeitungszeit bestimmter Aufgaben mit multimodalen Systemen. Der praktische Teil wird mit einem Vergleich von Memo und CogTool abgeschlossen bei dem die Werkzeuge zunächst einordnet werden, und dann anhand der Vorhersagegüte sowie subjektiv beurteilt werden.

Zusammenfassung und Ausblick kondensieren den Mehrwert dieser Doktorarbeit und zeigen Ansatzpunkte für die zukünftige Forschung auf.

Acknowledgements

I want to thank everybody from the Quality and Usability Lab at Telekom Innovation Laboratories and the research group prometei at Technische Universität Berlin for supporting me while working on this thesis. It was an honor for me to work together with so many kind and excellent people.

Special thanks I would like to address to my supervisor Prof. Sebastian Möller, who supported me all the time with help and advice, and gave me the chance to accomplish this work. Thanks for your patience, for the possibility to work with you, and for the pleasant atmosphere in your team. I also would like to thank Prof. Manfred Thüring, my co-supervisor, for helpful discussions about my research and for opening me the exciting field of cognitive modeling. Many thanks to Prof. Kristiina Jokinen for the interest in my work and the willingness to be available as reviewer of this thesis.

I want to give thanks to my mentor Robert Schleicher, you always ask the right questions at the right time, and with great flair you can drive discussions in the right direction. Ina Wechsung and Julia Seebode supported me not only professionally, but were and are still super buddies and just lovely people. I want to thank Matthias Rath and Sebastian Werk for their tutoring in math and Python, Nele Pape and Jeronimo Dzaack for their tutoring in cognitive modeling, and the whole MeMo group, including Stefan Hillman, Aaron Russ, Matthias Schulz and Klaus Engelbrecht for help and fruitful discussions. I thank Florian Metze for smoothing me the way into T-Labs and supporting me during my PhD as a short time scholar at the Interactive Systems Lab at Carnegie Mellon University. I also thank Michael Minge and Benjamin Jöckel who did great jobs in their master theses which I was allowed to supervise, and Arash Zandi and Max von Schlippe for their great support as student workers. Thank you Irene Hube-Achter, Yasmin Hillebrenner, Sandra Widera, and IT guys, your enthusiasm is enabling this whole circus.

I thank all researchers who make their work available free of charge.

I acknowledge the financial support by the DFG (Deutsche Forschungsgemeinschaft) who funded my research grant and the research group prometei.

I am deeply thankful to Sonja Eilmes, you are always there when I need you, and you manage both, to remind me that there are other things in life then multimodal

systems, and to give me the strength to carry on when I was sure there is no use in succeeding my work.

Finally, I am thankful to my family and friends for supporting me, hanging around with me, and having fun together. Live long and prosper and don't forget "I love you more!".

Contents

Acronyms

ACT-R	Adaptive Control of Thought-Rational
ANOVA	Analysis of Variance
ASR	Automatic Speech Recognition
AUE	Automated Usability Evaluation
AVP	Attribute-Value Pair
GOMS	Goals, Operators, Methods, and Selection Rules
GUI	Graphical User Interface
HCI	Human-Computer Interaction
LD	List depth
MDS	Multimodal Dialog System
MSA	Modality Selection Algorithm
OKS	Open Microphone in Combination with Key-Word-Spotting System
RBA	Restaurant Booking Application
SIMS	Sequential Independent Multimodal System
UI	User Interface
WIMP	Windows, Icons, Menus, Pointer

Chapter 1
Introduction

Recent developments in human-computer interaction (HCI) show that an increasing numbers of dialog systems offer more than one option to enable user input. Smartphones or navigational systems often come with an additional speech interface (Schaffer et al., 2016). Speech-based interactive systems are a subject of current research (Jokinen and Cheng, 2010) and new devices such as smart watches arise whereby speech as an input modality is becoming increasingly important (Nurminen et al., 2015). However, the Graphical User Interface (GUI) is still the more common input modality in those systems.

Touch screens are now a standard input method for the GUI. If an additional speech interface is integrated into a system where only a GUI was previously available, a multimodal dialog system (MDS) is built. Examples of such systems are Apple's iPhone extended with the speech interface Siri (Apple, 2011) and Android smart phones provided with google voice search (Franz et al., 2006). Both systems enable the use of selected spoken commands for specific interactions.

Employing speech input often saves interaction steps or time. A novice user deploying this benefit, however, will not know exactly if or how a speech-based interaction is possible. Experience is needed and, if the user in not accustomed to the system, reasoning and decision-making processes increase the cognitive load (Sweller, 1988). One possible way to avoid this additional load would be to make any modality possible at any point in the interaction.

Input modalities would then be processed sequentially and independently (Nigay and Coutaz, 1993). Sequential processing here means that users may perform system input only consecutively regardless which modality they use. Independent processing means that modalities are interpreted separately and no semantic fusion is performed. In doing so, a system would provide a graphical input element like a button and a speech input for each interaction. The previously mentioned smart phone examples fit only partly into this category of systems, as they integrate speech interaction only for very specialized tasks like menu navigation or keyboard typing.

In the domain of consumer products almost no sequential independent multimodal systems (SIMS) that allow any interaction using any modality have appeared on the market so far. An unsolved issue with speech input is that direct interaction

via spoken commands still does not work sufficiently well because actual automatic speech recognition (ASR) modules cause too many errors, e.g., by processing extraneous background noise as user input. Accordingly, push-to-talk or "Open microphone in combination with Key-word-spotting Systems" (OKS) are used to enable speech input. However, these implementations need at least one additional step during the interaction to activate ASR.

Assuming further improvements in ASR technology during the next years, future SIMS might have no need for push-to-talk or OKS. OKS might be implemented in an effective manner minimizing interaction time and cognitive resources. With these systems, any graphical and speech input will be entirely possible for each achievable task. Further, users will have to select input modalities in each step during the dialog with the system.

The selection of input modalities is influenced by various factors. Several studies revealed that input performance (like ASR error rate) and modality efficiency (measured in the number of turns to solve a task) are significant moderators in multimodal systems integrating a graphical and speech input (Möller et al., 2011; Schaffer et al., 2011a; Wechsung et al., 2010). Other documented influence factors are cognitive demand (Wickens and Hollands, 2000), interaction time (Bevan, 1995), hedonic quality (Hassenzahl et al., 2003), environment, and dynamic as well as static user attributes (Bohn et al., 2005; Ren et al., 2000).

The evaluation of the quality of MDS is a a current research area (Möller et al., 2011; Perakakis and Potamianos, 2008; Turunen et al., 2010). Thereby many methods are based on evaluations, using questionnaires in order to gain quality perceptions from real system users (Metze et al., 2009; Kühnel et al., 2010; Turunen et al., 2010). However, studies with real test subjects are at most expensive with respect to typically limited resources of time and money. Here effort could be saved by automated usability evaluation (AUE). In AUE predictions about quality factors of a system can be created automatically. The predictions are usually derived from parameters, which are obtained by the simulation of interaction between models of users and systems.

1.1 Research Gaps in multimodal Human-Computer Interaction

Even if quantitative data is collected in most of the studies about the choice of input modalities, the obtained results are often qualitative in nature. But even more harmful for the above mentioned purpose of modeling is the fact that, in most cases, the gathered quantitative data is not comparable due to differing experimental setups, interfaces, logged data, or user groups. This lack of quantitative data still keeps HCI researchers from a better understanding of the interdependencies of the factors influencing modality selection and complicates the construction of applicable theoretical and computational models. The existence of these models is a requirement for facilitating simulation of multimodal HCI between user and system models, which

further enables automated usability evaluation (AUE). Addressing this complex of problems this work identifies the following three research gaps:

1. The lack of quantitative findings regarding modality selection.
2. The lack of computational models for modality selection.
3. The lack of AUE tools for the simulation of multimodal HCI.

The first research gap addresses empirical research. In order to design empirical studies investigating modality selection the adoption of a cognitive perspective is important because the participants of experiments must be able to perceive the relevant information. Foundations of decision-making, human information processing, multiple ressource theory (MRT), and heuristics have to be adopted in order to appropriately consider the human factor (Thüring, 2002). It should be able to provide quantitative findings regarding modality selection, if the experimental setups are mostly uniform concerning interfaces, log data, and user groups. Further the independent factors have to be varied in a uniform manner and in appropriate levels. The above-mentioned factors input performance and modality efficiency turn out to be adequate candidates for a unified quantitative investigation of modality selection as they showed significant effects in previous studies (Möller et al., 2011; Schaffer et al., 2011a; Wechsung et al., 2010). However the mentioned studies investigated different systems, and further more different levels of the independent factors are needed for model creation. Therefore quantitative modality selection data has to be collected. In order to collect the amount of data that is needed for the creation of a model of modality selection that handles more then one influence factor a series of uniform experiments has to be conducted.

The second research gap addresses the field of computational modeling. If the first gap can be closed target data for computational models is available. Utility functions are often used for making rational decisions (Gray et al., 2006). Probabilistic (Möller et al., 2006) or cognitive models (Anderson et al., 1997) can utilize utility functions to simulate decisions for modality selection. Therefore the eventual aim is to compile a model utilizing a utility function for the prediction of modality usage in SIMS.

The third research gap addresses the practical application of modality selection within AUE tools. If the second gap can be closed an algorithm for modality selection is available. The AUE tools to be extended should already support both graphical and speech input. In addition, the tools should differ in their fields of application, so that the integration of the modality selection algorithm can be tested for these different areas. On the one hand MeMo (Möller et al., 2006) was identified. MeMo finds different interaction paths for a task and identifies potential usability issues. On the other hand CogTool (John et al., 2004) was identified predicting task execution times of skilled users. The eventual aim is to gain an understanding about the expected modality usage and the expected interaction steps during certain tasks executed with MeMo. Next the same tasks should be simulated with CogTool to predict the total multimodal task execution time.

1.2 Structure of this Thesis

In Chapter 2 theoretical foundations in the fields of multimodal interaction and automated usability evaluation are laid. The two AUE tools MeMo and CogTool are introduced. Further the research questions of this work are formulated.

Three experiments investigating the impact of modality efficiency and input performance on modality choice are presented in Chapter 3. It is shown how the quantitative data is merged into one database.

In Chapter 4 the computational model is created. An analysis of the predictive power of the model is conducted and an application demonstrates benefits and limitations of the developed modality selection mechanism.

Chapter 5 describes the integration of the modality selection algorithm into MeMo, and a procedure to render ACT-R models generated with CogTool multimodal. The integration of the modality selection algorithm is tested for both tools and application examples for the prediction of task steps and the prediction of total task execution time are given. The chapter closes with an comparison of the tools.

Chapter 6 summarizes the work and gives indications for future research.

Chapter 2
Foundations

In this chapter theoretical foundations in the fields of multimodal interaction and automatic usability evaluation are laid. Basis knowledge about the concepts and terminologies of multimodal human computer interaction and the choice of modalities is essential, not just to understand the factors influencing the modality selection, but also to vary them effectively in quantitative studies. The first research gap studied in this work is uncovered, namely the lack of quantitative findings regarding modality selection. Theories and models of human decision-making are usually based on empirical evidence. Empirical studies investigating human modality selection behavior provide a basis for testing theories and new models. An application of such theories as computational models can be found in the field of automated usability evaluation. This topic is the subject of the second research gap of this work, which is the formulation of a computational model for modality selection. The lack of such a model prevents the automatic usability evaluation of multimodal HCI. Therefore the third research gap of this work is that no automatic usability evaluation tools for the simulation of multimodal HCI exist. Two tools, namely the MeMo workbench and CogTool, are presented.

Theoretical foundations of multimodal interaction are presented in Section 2.1. Section 2.2 provides the basics and state of the art of automated usability evaluation. The MeMo workbench in introduced in Section 2.3 and CogTool in Section 2.4. In Section 2.5 research questions are derived from the identified research gaps. Section 2.6 concludes with a summary of this chapter.

2.1 Multimodal Interaction

The concepts and terminology used in this work are described in Section 2.1.1. At first the term human computer interaction is described. After the concept and the usage of the term modality have been introduced, insights into the field of multimodal human computer interaction are given. Section 2.1.2 provides theories and principles about human modality choice that has to be considered during the design

of multimodal systems and during the planning of interaction experiments. Further in Section 2.1.3 the influencing factors input performance and modality efficiency are introduced.

2.1.1 Concepts and Terminology

2.1.1.1 Human-Computer Interaction

The interdisciplinary research field of human-computer interaction (HCI) connects findings of computer science, psychology, work science, cognitive science, ergonomics, sociology and design to develop novel interfaces between users and computers. HCI researchers observe the ways users exchange information with computers (Charwat, 1992). The term human computer system is used, if a person or a group of persons are interacting with a computer to perform a specific task (Timpe and Kolrep, 2002). Human-computer systems always have a feedback structure as controlling or regulating actions of users influence the state and therefore the feedback from the computer. In human-computer systems the mutual exchange of information takes place using a user interface. The user interface provides information about the state of the computer in a way that is perceptible for humans and allows to make inputs to its technical process.

Nowadays the user interface is seen as a key element in the provision of information in human-computer systems, therefore, a good design is of particular importance (Streitz, 1988). The quality of task performance is largely determined by the usability of the interface, which has to be evaluated under consideration of knowledge and skills of the users, and limiting context-dependent factors. ISO9241-210 (2009) defines usability as

> "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use."

Thus, usability is influenced by the following criteria:

- **Effectiveness**: the completeness and accuracy with which users achieve specified goals (often measured as degree of task completion).
- **Efficiency**: the resources expended in relation to the accuracy and completeness with which users achieve goals.
- **Satisfaction**: the subjective assessment of the participants on a validated questionnaire.

In this context effective human-computer systems require a user interface that constantly ensures a smooth two-way exchange of information. Since the human users are equipped with a variety of specialized interaction possibilities, the exchange of information can take place on different channels. The basic idea of multimodal systems aims at realizing and utilizing the greatest possible range of human

interaction possibilities for information input and output. The technical implementation of such a channel is called ”modality”. A definition of the term and the relationship between human sensory perception and technical feasibility in the design of user interfaces are described in the following section.

2.1.1.2 Modality

Gibbon et al. (2000) state that a confusion in terminologies exists in HCI and argue that the lack of agreement can probably be attributed to the interdisciplinary nature of the field. Whereas in the technical sense the term modality refers to the concrete combination of an interaction device with an interaction language (Engesser and Claus, 1993), in a physiological sense it is understood as the possibilities of human perception (sensors) and human action (motor). Generally, the term ”modality” can be understood as the specific way how certain information between a sender and a receiver is provided or exchanged. In this respect a modality refers to a communicative system, which is characterized by the way in which information is encoded and interpreted (Beuter, 2007).

Hedicke (2000) distinguishes between the so-called action modalities and perception modalities. Action modalities refer to the available input forms to transmit information from the user to the system, whereas perception modalities relate to the transmission of information from the system to the user. To simplify some authors refer perception modalities as output modalities and action modalities as input modalities (Engesser and Claus, 1993). According to Charwat (1992) currently the three modalities visual, auditive, and tactile (referring to the three senses sight, hearing, and touch) are most relevant for HCI, whereas in physiology at least three more senses, namely smell, vestibular, and taste, are defined.

In fact, the largest proportion of information transmission is so far using the visual modality (Norman, 1990). Usually electromagnetic radiation of different wavelengths are sent via monitors, displays, diodes, etc. Users perceive the visible radiation as light and process the information to the characteristics brightness, color, size, shape, orientation, distance, movement and direction (Muthig, 1999). The auditory perception modality is utilized by sending information in the form of sound waves by means of loudspeakers to the user. This information can be perceived and processed as non-verbal sounds or sound sequences, as meaningful sounds or verbal speech information. Especially in the field of mobile applications tactile stimuli get increasingly important. Using motors or actuators certain forces are generated to contact the user by means of vibration or pressure.

Action modalities describe how information is registered by a system and in what way the user can interfere. In the simplest case, action modalities can be classified analogous to perception channels of the user. In this context Hedicke (2000) distinguishes between an auditory, visual and haptic user interface (see Figure 2.1).

While the auditory interface for inputting information records verbal (e.g. speech input) or non-verbal (e.g. clapping of hands) sounds from the environment via a microphone, visual interfaces recognize information like gestures and facial expres-

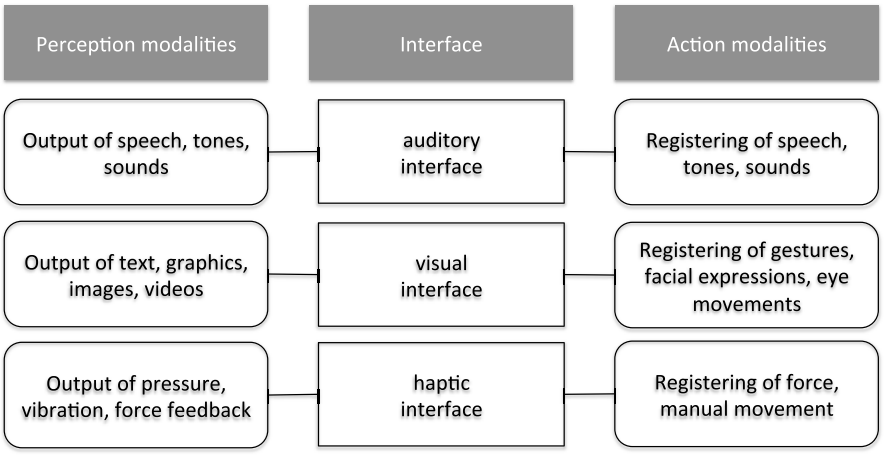

Fig. 2.1 Action modalities classified analogous to perception channels of the user.

sions of the user by means of camera systems. Haptic interfaces detect information from state changes of parts of a device (e.g. when pressing buttons, operating controls, when moving units or parts or during deformation of units).

With a few exceptions (Iwata et al., 2004) perception modalities for the vestibular, olfactory, or gustatory sensory system so far play a minor part in the design of human-computer systems. With regard to the action modalities other ways to utilize explicit and implicit information about the user for system input are part of current research. Current research topics can be located in the fields of brain-computer interfaces (Wolpaw and Wolpaw, 2012), emotion detection (Mahlke and Minge, 2008) and mental workload (Nickel et al., 2002).

As this section has shown, there are many opportunities in HCI to exchange information on both sides the perception side and the action side. As the most important modalities were named and described, the following section is dedicated to the combined use of modalities, introducing the term multimodality. In the following sections, the terms input modality and output modality are being used.

2.1.1.3 Multimodal Human-Computer Interaction

Considering the term multimodal as addressing different sensory channels simultaneously for information transfer, it is often associated with the output (or perceptual) modalities of users (Weidenmann, 1995). However, many efforts in the field of multimodal HCI aim at realizing a wide range of parallel available input (or action) modalities, allowing users various options to enter information. A frequently used definition for multimodal systems provides Oviatt (2003):

> ”Multimodal systems process two or more combined user input modes – such as speech, pen, touch, manual gestures, gaze, and head and body movements – in a coordinated manner with multimedia system output.”

The term multimodal is here related to the input modalities, while the systems output modalities are referred to as multi-media. According to Oviatt’s definition a system comprising at least two input modalities and one or more output modalities for information exchange can be understood as multimodal. Wechsung et al. (2012) define multimodal dialogue systems as:

> ”computer systems with which human users interact on a turn-by-turn basis, using several different modalities for information input and/or receiving information from the system in different modalities.”

In this thesis the term multimodal system is used as defined by Wechsung et al. (2012), as it fits better to the understanding of input modalities and output modalities described in Section 2.1.1.2.

In order to ensure the required high usability of the user interface, the interplay of input and output modalities must be designed in such a way that the system can effectively and efficiently meet its external or self-imposed objectives (Nigay et al., 1995). Addressing this issue Nigay and Coutaz (1993) developed a design space for multimodal systems along three dimensions. The first dimension, the *level of abstraction*, defines the technical level, on which information of the different input and output devices is processed (e.g. speech input may be processed as a signal, a sequence of phonemes or as meaningful parsed sentences). The design space considers the two values ”Meaning” and ”No meaning”. The second dimension, the *usage of modalities*, specifies the temporal availability of the modalities. The design space considers parallel or sequential usage. The third dimension referres to the *fusion* of modalities and describes if and how the information of the different modalities is combined. The design space here considers the classes ”independent” (no fusion) and ”combined” (fusion is implemented).

Based on these dimensions four different classes of multimodal systems can be identified (each of the classes can have two different levels of abstraction):

- **Exclusive**: modalities are used sequentially and independently (fusion is not implemented).
- **Alternate**: modalities are used sequentially. Fusion, the combination of input information, is possible.
- **Concurrent**: modalities can be used in parallel, but are processed independently (fusion is not implemented).
- **Synergistic**: modalities can be used in parallel. Fusion, the combination of input information, is possible.

Together with this design space Nigay and Coutaz (1993) provide a classification scheme that specifies the location of a system within the design space. This classification can be used for a structured comparison of different implementations of a multimodal system through usability testing. The system used in the presented studies offers sequential and independent (no fusion) input using a touchscreen-based

graphical user interface (touch input) or a speech-based user interface (speech input). It can therefore be classified as exclusive in the design space of Nigay and Coutaz (1993).

In order to guide the assessment and evaluation of multimodal services, Möller et al. (2009) developed a taxonomy describing quality aspects of multimodal interaction. Wechsung et al. (2012) and Wechsung (2014) further elaborated the taxonomy. On the user side assumptions of the Multiple Resources Theory by Wickens (2002) sketched in Section 2.1.2.3 are adopted. On the system side the taxonomy describes several aspects related to interaction performance. One aspect is input performance, which is particular important for this work as it is one of the explored moderators of modality selection. According to Wechsung et al. (2012) a modalities input performance can be quantified e.g. in terms of accuracy or error rate, as it is common practice for speech, gesture recognizers and facial expression recognizers. In Section 2.1.3 it is outlined how such an error rate can be calculated, as input performance is quantified in terms of error rate in this work.

Another important aspect that is also part of the taxonomy is efficiency. In the context of multimodal interaction, Perakakis and Potamianos (2008) define modality efficiency on the basis of the time required by that modality to complete a task. However, in a number of studies increased speech usage could be observed, if speech input offered a shortcut in terms of a reduced number of necessary interaction steps. Wechsung (2014) summarized exemplary studies where speech input was preferred for selecting an element out of a very long list (Raisamo, 1999), for entering a long telephone number (Naumann et al., 2008), or for searching for specific titles (Naumann et al., 2009, 2008; Metze et al., 2009). For entering names in a query system, Rudnicky (1993) observed that speech was the preferred modality compared to keyboard and a scrolling bar, whereas speech input was less efficient in terms of task-completion time. The explanation of this opposing result might be that entering a name requires only one interaction step via speech but usually more then one interaction steps via keyboard or a scrolling bar. For these tasks speech input is containing a shortcut in terms of interaction steps, it can therefore be assumed that interaction steps are a more appropriate measure for modality efficiency. As the empirical studies presented in this work contain a list browsing task, modality efficiency is measured in interaction steps.

Known HCI principles for multimodal interaction provide useful guidance for designing multimodal interfaces. Synergy, robustness, modularity, customizability and consistency, including symmetric multimodality (Wahlster, 2003), are important features of successful multimodal dialogue systems and design tools (Potamianos and Perakakis, 2008). Further the approach of universal accessibility aims to produce systems that can be used by everyone in every context, permitting the user to exercise selection and control over how they interact with the computer (Obrenovic et al., 2007; Oviatt, 2003). The selection is then guided by individual abilities and preference, as well as the usage context.

As described above the selection of input modalities plays an important role in the area of universal multimodal systems. Therefore the next section will provide insights in the field of human decision-making.

2.1.2 Choice of Modalities

How people make decisions and what mechanisms play a role is of interest for many disciplines. The development and the empirical validation of decision theories has a long tradition, especially in the economic, social, and political sciences. Since the human factor has gained attention, insights from decision research are increasingly taken into account in the engineering of human-computer systems (Wickens and Hollands, 2000).

2.1.2.1 Characteristics of Decisions

Decisions describe the choice between at least two options or alternatives based on personal preferences. Some characteristics of decisions include whether a problem with binary or multiple alternatives exists, whether it is a single or an iterative decision-making process or whether decisions are made by an individual or by a group (Duffy, 1993).

Decisions are always made with regard to the consequences of an option (Tversky and Kahneman, 1992). Another important characteristic of decisions is the degree of uncertainty about the possible consequences, since the relationship between an option and a consequence often have a probabilistic nature. In decision-making research decisions are called *under risk* if the probabilities of the possible consequences are known and *under uncertainty* if the probabilities are not known (Jungermann et al., 1998).

The decision when buying a car if two types of cars are available represents, for example, a comparably safe decision, if preliminary information on the advantages and disadvantages of both vehicles has been gathered. By means of a multiattributive cost-benefit analysis, the individual characteristics of each car can be separately evaluated, weighted according to personal preferences, and then summed for the two available options. The results are total values; on that basis a decision can be made (Wickens and Hollands, 2000).

In many decision-making situations, especially in dealing with complex, dynamic technical systems, however, either the consequences or the probability of their occurrence are unknown. Decisions are ”risky” if some of the possible, but uncertain outcomes are particularly unpleasant or associated with high costs (Tversky and Kahneman, 1992). For a complex, uncertain decision problem Wickens and Hollands (2000) give the example of a pilot deciding to continue his flight in unsafe weather or to turn back. In this case, the consequences are probabilistic, meaning that it is difficult to predict the impact the weather will actually have on the safety of the flight.

2.1.2.2 Types of Decisions

According to Wickens and Hollands (2000) three types of decisions can be distinguished: rational, cognitive and naturalistic decision-making.

- The rational or normative decision-making research examines how people make decisions corresponding to an optimal framework, or a ”golden standard”, maximizing profits and minimizing losses.
- The cognitive or information-processing decision-making research considers the extent to which errors or distortions in the decision-making process can be attributed to limited human attention, working memory or selection strategies and familiar decision routines.
- Naturalistic decision-making research is interested in the decisions in real applications, taking into account significant real world factors, such as domain-specific expertise, time constraints, environmental dynamics, high risks and trade-offs (Zsambok and Klein, 2014).

The rational decision-making research provides the briefly sketched multi-attributive cost-benefit analysis, a rational decision-founded instrument. However, the rational approach often turns out to have only limited validity compared to naturalistic decision-making. For example, people do not always seek to maximize benefits or minimize losses, and there are differences between objective and subjective values and probability estimates, leading to distortions of judgments, which can be explained partly by certain cognitive decision routines (heuristics).

Handling the empirical research presented in this book requires the consideration of human information processing, and therefore the perspective of cognitive decision-making has to be considered. On the other hand, the generation of a model for the prediction of modality usage will require the application of a rational decision-making approach, allowing the computational processing of the investigated factors. In the following section, basics of human multimodal information processing will be outlined. Two approaches allowing for computational processing will be presented in Section 2.3 (MeMo) and 2.4 (CogTool).

2.1.2.3 Multimodal Information Processing

Aiming at a better understanding of multimodal HCI first of all human information processing in general has to be considered. According to the model of human information processing by Wickens and Hollands (2000) selective attention, diagnosis, and response selection can be considered as the main stages of a decision. As a first step selective attention filters information perceived from the environment. Only those stimuli, which are considered to be currently relevant for the situation due to the expertise of the decision maker are forwarded for perceptual processing. Next a diagnose, as an understanding or an assessment of the decision situation is derived from the filtered information. Thereby information from longterm memory, situation awareness (Endsley, 1995), and meta-cognition (Reder, 1988) are inte-

grated to derive a general understanding about the system. In the next stage, based on the diagnosis, the process of selection of an action in relation to the expected consequences and the associated values of a decision (cost benefit analysis) is initiated, which again triggers the action execution. According to this model it is crucial that human beings are aware of this processing, and that processing resources are limited, as well as susceptible for interferences. As interferences can impair the information exchange in HCI, a successful implementation of multimodal interaction requires that interferences are impossible or at least minimized.

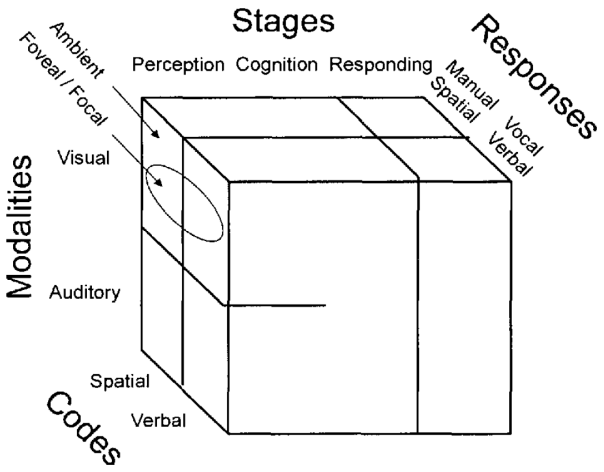

Fig. 2.2 Dimensions of the multiple resource theory (MRT)

In oder to identify interferences at an early stage of development and to appropriately coordinate modalities in a human-computer system, the multiple resource theory (MRT) of Wickens and Hollands (2000) can be employed. The basic idea is that human users have different resources for the simultaneous execution of cognitive processes. As a consequence, in dual task situations, individual tasks can be processed at the same time, if the two tasks use different resources. As depicted in Figure 2.2 MRT proposes three different processing stages, two different response codes, two different perceptual modalities and two different input codes. Input codes are distinguished for spatial or verbal perception of the human user. Correspondingly the response codes are distinguished as manual or verbal actions. The processing stages split into perception and cognition, as well as responding, whereas MRT assumes one pool of resources for the first two, and another one for response selection and response execution. On the part of modalities the authors distinguish between auditory and visual resources. A detailed description can be found in Wickens and Hollands (2000). In (Wechsung, 2014) the assumptions of MRT were adopted, and further aligned for multimodal systems. Regarding the perceptual modalities additionally the haptic modality is suggested. Accordingly haptic

responses (e.g. responses including touching and moving the system, like manual responses) are added to the response codes.

From a cognitive psychology point of view, the model of human information processing and MRT describe how humans acquire multimodal information from a human-computer system and how human decisions are made on the basis of present knowledge. Thus, a decision to use a specific modality, if equivalent modalities are offered, can be made on the basis of selectively perceived and interpreted cues. As stated in Section 2.1.2.1 in many decision situations not all relevant information is available or the best decision can not be determined by a retrieval of information from long-term memory (Tversky et al., 1990). Nevertheless, humans are able to make decisions in situations where the consequences are associated with uncertainty. The strategies that are applied in such cases are called heuristics. The most important heuristics for human decision-making are:

Anchoring and adjustment: the heuristic describes the tendency to rely too heavily on a rough first hypothesis (the ”anchor”) when making decisions.

For input modality choice the anchoring heuristic can cause users to quickly make a first hypothesis about the usefulness of the available modalities, and cause them to a repeated use of a particular modality. The intentional change to another modality could be more difficult, because people hold on to their first decision, thus avoiding the cost of a modality change, which is causing cognitive effort.

Availability heuristic: the heuristic describes the ease with which information can be retrieved from long-term memory – the lower the cognitive effort of an action, the more likely the selection of this action.

This implies that users need to have initial experience with the use of input modalities, in order to easily imagine the use of these modalities in future situations. If negative experiences occur when using a specific input modality (e.g. ASR errors), the availability heuristic may cause that this modality is used less frequently in the future.

Representativeness heuristic: the heuristic describes how similar a perceived element is to the abstract model stored in long-term memory.

In multimodal HCI a user can for example experience that certain system inputs can be made faster and easier using a specific input modality. In future situations when alternative modalities are offered, the use of this modality will appear particularly representative.

As sketched above the human decisions are guided by a number of cues. The decision for a specific input modality is also guided by such cues. Therefore the next section will provide insights in factors influencing the choice of input modalities.

2.1.3 Influencing Factors of Modality Selection

Several factors influencing modality selection were already mentioned. A detailed description of known factors should help to identify factors that are suitable for a computational model.

2.1.3.1 Input Performance

In HCI, effectiveness is defined as " the completeness and accuracy with which users achieve specified goals" (ISO9241-11, 1998). In interactive systems, effectiveness is essentially affected by the input performance (error-proneness) of the interface (Card et al., 1990). Bilici et al. (2000) tested a multimodal setup, where participants switched from speech to manual input when ASR errors arose. Suhm et al. (1999) further found that users tend to select the less error-prone modality after repeated usage of a multimodal system. Effects of input performance are not well documented for graphical user interfaces. In a study of our own we observed that participants strongly preferred the speech input if touch input errors occurred (Schaffer and Minge, 2012). Input performance is typically quantified in terms of error rates. The word error rate (WER) is the sum of transcription errors (word substitutions, deletions, and insertions), divided by the number of reference words with lower scores indicating better performance (Möller et al., 2011). Actual ASR systems can widely differ in WER, reaching e.g., from 4.1 % for command and control operations in mobile phones (Varga et al., 2002) to 46.7 % for speech-to-text transcription of conference room meetings (Fiscus et al., 2008). For AUE of multimodal systems this variance means that a wide margin of error rates has to be considered in order to forecast modality choice. These large variances result from different underlying conditions such as acoustic conditions or the size of vocabulary and grammar. Metrics comparable to WER can also be calculated for other input modes like touch screen (Kühnel et al., 2010).

2.1.3.2 Modality Efficiency

As already mentioned in Section 2.1.1.3 this work uses the term modality efficiency which is measured in interaction steps. Efficiency is defined as the effort expended by the user in relation to the accuracy and completeness of goals achieved (ISO9241-11, 1998). High efficiency is reached when the user fulfills a task while expending as few resources as possible. For HCI, different metrics can be used to assess effort, such as task completion time, monetary costs, mental effort of the user, or the number of interaction steps required to solve the task (Bevan, 1995). Duration-related metrics are often used to assess the efficiency of multimodal systems. Perakakis and Potamianos (2008) showed that speech input usage increases, if it is more efficient compared to a GUI, measured in overall time spent in a modality. On the other hand, users tend to use modalities that save additional or inconvenient

interaction steps, even if interaction time increases (Rudnicky, 1993). Similar results were found by Wechsung et al. (2010) and in our own studies presented in Chapter 3. .

2.1.3.3 Mental Effort

Mental effort can be described as an operator's attentional capacity in relation to environmental demands (Kahneman, 1973). The Multiple Resource Theory (Wickens and Hollands, 2000) introduces a model of attentional capacities to describe mental workload. High mental workload can be caused by multiple tasks accessing identical cognitive resources. Task fulfillment can be improved if perception and information processing is allocated to distinct resources. Workload interferences between tasks and allocation capabilities can be identified by means of the model. In a SIMS study, Schaffer et al. (2011a) showed that perceived mental effort increases with increasing task complexity for both speech and touch input. A follow-up study revealed that participants select the modality that reduces task complexity if one of the modalities is more efficient in terms of mental effort (Schaffer et al., 2011b). In summary, participants were able to keep perceived mental effort constant by employing specific modalities.

2.1.3.4 Hedonic Quality

If a product not only satisfies the task-related requirements, but also generates positive feelings in the user, it has hedonic quality (Jordan, 2002). In contrast to pragmatic quality, which focuses on efficient and effective goal achievement, hedonic quality asks for novel interaction techniques and communicates a desired identity through a professional, cool, modern, or different appearance (Hassenzahl et al., 2003). For multimodal HCI, users' modality choice can be affected by the perceived innovative energy or originality attributed to a modality, with innovative modalities being used more frequently.

2.1.3.5 Personal Preferences

Personal preferences thus can have a decisive influence on individual modality usage profiles. It was observed that some users did not switch modalities at all but used only either a touch or speech input (Schaffer et al., 2011a). Further user attributes like physical impairment, attitude, character traits, education, expertise, and affinities can have an influence (Bohn et al., 2005). Especially when dealing with mobile devices, the situational context (Dey and Häkkilä, 2008) and aspects like lighting conditions, surrounding sounds, temperature, legal requirements, and social desirability also have to be considered to explain user behavior (Bernsen, 2008).

2.1.3.6 Implication

Overall, it must be observed that all considered factors are interrelated. Because personal preferences change dynamically, they can only be forecasted if information about previous behavior is available. Furthermore, the influence of static user attributes and situational context on modality choice so far has not been examined sufficiently enough to extract computational models. Another constraint relevant to most of the reviewed studies is that not all findings can be generalized and thus cannot usually be compared directly. The conclusions drawn are always post-hoc explanations of user studies with differing material, setup, and participants that have an uncontrolled impact on the various factors influencing modality choice. The need for stable experimental conditions to gain deeper insights for specific factors has motivated our series of experiments. In the consequence of a rational decision-making approach, factors related to efficiency and effectiveness, such as input performance and modality efficiency should be taken into account to forecast average user behavior. The eventual goal of this work is to make the findings about the influence factors available within AUE tools. Therefore, basics of AUE are presented in the next Section.

2.2 Automated Usability Evaluation

Section 2.2.1 provides a classification of AUE within Nielsens usability engineering lifecycle and a general motivation for conducting research in this area. First insights into the state of the art of AUE are presented in Section 2.2.2. In Section 2.2.3 developments in multimodal AUE are outlined. Approaches for the comparison of simulation approaches are sketched in Section 2.2.4.

2.2.1 Classification and Motivation

Figure 2.3 shows Nielsens usability engineering lifecycle model with its stages analysis, design, prototyping, expert evaluation, empirical testing, iterative design, and feedback from field (Nielsen, 1993). For each of the stages specific methods can be applied to consider user needs during the process of developing a user interface. If the behavior of real users while interacting with a newly designed interface is of interest, usually at least a prototype has to be available. In the lifecycle model the first stage where real user behavior can be observed is empirical testing, e.g. by using the think-aloud method (Lewis and Mack, 1982). In earlier stages user interface (UI) designs can be evaluated by applying e.g. cognitive walkthroughs (Wharton et al., 1994) or a heuristic evaluation (Nielsen, 1994). Compared to these methods AUE can be employed in earlier phases of the lifecycle. Simulations can be performed with first prototypes before the real system is available. Usability errors in the designs can early be identified and eliminated before the first implementations of the real user interface.

Regarding multimodality this also means that several designs integrating different modalities can be tested against each other. Thereby the system designer can optimize the interaction design of the planned system aiming at finding a combination of modalities maximizing efficiency and effectiveness of the interaction.

For a classical usability test usually a prototype has to be available in order to enable the collection of interaction and behavior data of system users. The typical procedure causes extensive effort including recruitment of test participants, as well as planning, execution and evaluation of the study. Nonetheless so far subjective methods are still mostly being used if a usability evaluation is performed, as most AUE methods are still only implemented as research tools.

Ongoing research efforts are also present in the field of subjective evaluation of the quality of MDS (Möller et al., 2011; Perakakis and Potamianos, 2008; Turunen et al., 2010). Judgments on quality are usually gained from user perception. Using questionnaires, users are asked to rate quality features of the system. The SASSI questionnaire was developed to assess subjectively perceived quality of spoken dialog systems (Hone and Graham, 2000). SASSI and other questionnaires were adapted for MDS (Metze et al., 2009) and new questionnaires are currently being developed (Kühnel et al., 2010; Turunen et al., 2010). Several studies have been carried out in this field in recent years. However, studies with real test subjects are

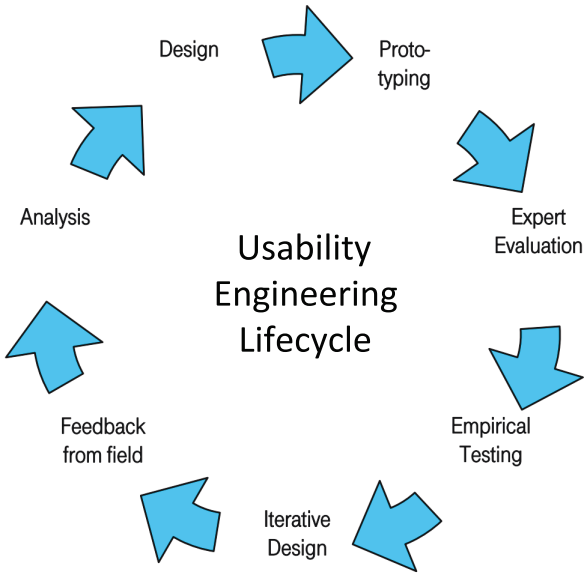

Fig. 2.3 The Usability Engineering Lifecycle as defined by Jakob Nielsens.

quite expensive with respect to typically limited resources of time and money. Here also in the field of multimodal HCI effort could be saved by automated usability evaluation (AUE), which is therefore a research topic.

2.2.2 State of the art in automated usability evaluation tools

AUE deploys computational modeling for task solving and decision-making in a simulated user. Computational models can integrate probabilistic behavior and rules (Möller et al., 2006) as well as utility functions (Gray et al., 2006) for making rational decisions. Cognitive models usually build up on cognitive architectures (Anderson et al., 1997), taking into account human attention, working memory, and strategy selection during decision-making. Also utility functions are often utilized during simulation of human cognitive processes (Fu and Gray, 2006). In AUE decisions like ”how to initialize the next input into the system” have to be made at each step by a simulated user.

Employing task knowledge, information about the user, and system specifications, the AUE simulation realizes interaction between user and system models. Interaction is simulated by means of tools like the MeMo workbench (probabilistic-rule based modeling) or CogTool (cognitive modeling) and interaction parameters are extracted by gathering log data during the simulated human-machine dialog

(Möller et al., 2006; John et al., 2004). A usability profile is generated at the end of the AUE process.

So far no solutions for MDS have left the research phase. One issue is that it is unclear how the simulated user should behave if more than one input modality is offered. Concrete knowledge about users' modality choice strategies is missing. This knowledge is needed to build and validate both theoretical and computational models. As only a small quantity of such data has been collected so far, this step has only rarely been taken. AUE simulation approaches and recent developments in the multimodal domain are discussed in the next section.

2.2.3 Developments in Automated Usability Evaluation of Multimodal Systems

An overview of usability evaluation automation (Ivory and Hearst, 2001) summarized 10 simulation approaches supporting automatic usability analysis: 9 cognitive architectures and 1 statistical modeling technique. Morrison (2003) tested the applicability of 19 computational human behavior representations for military simulations, including 7 of the cognitive architectures discussed by Ivory and Hearst (2001). Both papers conclude that the level of expertise required to successfully deploy cognitive modeling is an impediment. Therefore the applicability of cognitive architectures for usability evaluation is still limited due to considerable learning time and the effort needed to employ them. Nevertheless, efforts are being made to utilize cognitive models for AUE. Kieras et al. (1997) predicted human performance in telephone operator tasks with models constructed in the EPIC architecture for human information processing. A simulated operator had to act in multiple modalities: handling a telephone system and processing customer requests. However, the sequence of modalities was predetermined by the task and modalities could not be selected. Anderson et al. (1997) developed the cognitive architecture ACT-R, which is used to model various psychological aspects, including strategy selection. Schaffer and Reitter (2012) produced a model, learning about the efficiency of multiple modalities while interacting with a SIMS. However, the implemented cognitive processes have not yet been validated. Further developments have tried to utilize cognitive modeling for AUE tools by integrating simplified models for specific domains. Distract-R allows designers to prototype new in-vehicle interfaces and to evaluate the specifications by generating predictions of driver distraction (Salvucci, 2009). CogTool is a user interface prototyping tool that automatically evaluates the design with a predictive human performance model to assess total task time (John et al., 2004; Bellamy et al., 2011). Basic modeling of SIMS is possible with CogTool and Distract-R. However, single task steps as well as modality changes have to be specified manually by the modeler.

The statistical simulation method discussed by Ivory and Hearst (2001) is AMME (automatic mental model evaluator), developed by Rauterberg (1996). Petri nets (Reisig and Rozenberg, 1998) are employed to reconstruct and analyze the user's

problem-solving process. In addition to a system description, AMME requires log file data in order to generate the Petri net. Although the Petri net is applicable for subsequent simulations, reusability for other systems cannot be assumed as interaction parameters and system description may not be reusable. Möller et al. (2006) presented the MeMo Workbench as a tool for semi-automated evaluation of interactive systems. The approach, supporting the design process as well as the evaluation of the design, is based on simulation with user models and system models and is suitable for simulations of speech or GUI-based interaction including touch screen (Engelbrecht et al., 2008). MeMo user models build upon probabilistic simulation according to data-driven, pragmatic, or theory-driven solutions. So far the workbench does not support the simulation of multimodal interaction. An approach by Schleicher and Wechsung (2012) predicts "later" modality preference, based on interaction parameters and quality ratings of the component modalities. As the method builds on ratings of perceived quality (available only if a prototype already exists) the applicability for simulation is limited. Although the authors report that models fit well, they conclude that results have to be validated with larger samples.

Efforts within the computational modeling community show that a high value is assigned to AUE. As future dialog systems will increasingly rely on multimodal input, multiple modalities will also have to be considered during the simulation of interactions. The utility of single modalities thereby plays a central role for SIMSs. In engineering psychology decision makers often choose the option with the greatest expected value (Wickens and Hollands, 2000). Cost-benefit considerations are employed in cognitive decision-making, e.g., for the selection of interactive behavior (Gray et al., 2006) as well as in rational decision-making (Sheridan and Parasuraman, 2000). In conclusion, both approaches rely on a rational analysis perspective, maximizing expected utility. Predictions are based on the assumption that human beings act similar to naive statisticians (Kahneman, 2011).

2.2.4 Comparison of Simulation Approaches

Balbo (1995) carried out a survey of usability evaluation automation, using a taxonomy that distinguishes among the following four approaches to automation: nonautomatic, automatic capture, automatic analysis, and automatic critic. Taking up this taxonomy Ivory and Hearst (2001) adapted the automation types to specify which aspect of a usability evaluation method is automated. The following levels are determined:

- None: no level of automation supported (i.e., evaluator performs all aspects of the evaluation method);
- Capture: software automatically records usability data (e.g., logging interface usage);
- Analysis: software automatically identifies potential usability problems; and
- Critic: software automates analysis and suggests improvements.

In their survey on the state of automation in usability evaluation Ivory and Hearst (2001) grouped usability evaluation methods along the following four dimensions:

- Method Class: describes the type of evaluation conducted at a high level (e.g., usability testing or simulation);
- Method Type: describes how the evaluation is conducted within a method class, such as thinking-aloud protocol (usability testing class) or information processor modeling (simulation class);
- Automation Type: describes the evaluation aspect that is automated (e.g., capture, analysis, or critique); and
- Effort Level: describes the type of effort required to execute the method (e.g., model development or interface usage)

In their taxonomy Ivory and Hearst (2001) classify five method classes. To these method classes different method types can be assigned. Table 2.1 mentions a few examples.

Table 2.1 Method classes and method types as described by Ivory & Hearst (2001).

Method class	Description	Method type (examples)
Testing	An evaluator observes users interacting with an interface (i.e., completingtasks) to determine usability problems.	Thinking-Aloud Protocol, Log File Analysis
Inspection	An evaluator uses a set of criteria or heuristics to identify potential usability problems in an interface.	Cognitive Walkthrough, Heuristic Evaluation
Inquiry	Users provide feedback on an interface via interviews, surveys, and the like.	Interviews, Questionnaires
Analytical Modeling	An evaluator employs user and interface models to generate usability predictions.	GOMS Analysis, Cognitive Task Analysis
Simulation	An evaluator employs user and interface models to mimic a user interacting with an interface and report the results of this interaction (e.g., simulated activities, errors, and other quantitative measures).	Information Proc. Modeling, Petri Net Modeling

The method types are determined by the used evaluation methods within the testing, inspection, inquiry, analytical modeling, and simulation classes. Referring to the classification scheme presented in Table 2.1 the tools used in this work can be located in the simulation class. The classification of the tools is shown in Section 5.3.

Ivory and Hearst (2001) also expanded Balbo's automation taxonomy by an attribute called effort level indicating the human effort required for method execution. The levels are:

- Minimal Effort: does not require interface usage or modeling.

- Model Development: requires the evaluator to develop a UI model and/or a user model in order to employ the method.
- Informal Use: requires completion of freely chosen tasks (i.e., unconstrained use by a user or evaluator).
- Formal Use: requires completion of specially selected tasks (i.e., constrained use by a user or evaluator).

The authors further mention that these levels are not necessarily ordered by the amount of effort required, since this depends on the method employed (Ivory and Hearst, 2001).

Taken as whole the classification scheme provided by Ivory and Hearst (2001) is useful to identify to which extent specific tools are applicable in the AUE context. However, since it is difficult to present special features of individual tools in a comparable way, these details are largely abstracted in the investigation of Ivory and Hearst (2001). Being aware of this limitation Ivory and Hearst (2001) also include subjective assessments for the discussed techniques using the criteria:

- Effectiveness: how well a method discovers usability problems,
- Ease of use: how easy a method is to employ,
- Ease of learning: how easy a method is to learn, and
- Applicability: how widely applicable a method is to WIMP (windows, icons, menus, pointer) and/or web UIs other than to those originally applied.

As a part of the software development process AUE tools should smoothly integrate into the usability engineering lifecycle. Therefore it is important that the creation of models in the AUE process is implemented in an easy to use, and reusable manner, and that the designs can be easily taken along into later stages of development. This concerns the system models as well as integrated user models and task models. Thereby it is taken into account that the modeler usually is not a programmer. This may affect the applicability of AUE tools in practice.

Another possibility for the comparison of AUE approaches is the quality of the predictions. If human data for the modeled task that can be compared to the simulation results of the employed tools is available the prediction performance of the tools can be assessed. Commonly used goodness of fit measures are R^2 (the coefficient of determination) and $RMSE$ (root-mean-square error).

2.2.5 Implication

Within this work a utility-driven model for modality choice is developed. In order to be able to create this model the lack of empirical findings regarding modality selection has to be overcome. Once the model is there it can be utilized within existing AUE tools in order to enable simulations of multimodal interaction. The tools to be extended should fulfill a few important requirements. The modeler should be supported in his work with easy to use graphical tools to create, save and modify

the necessary models. This also implies that programming should not be necessary for the modeler. Graphical and speech input should already be supported, in order to lower the implementation overhead during the integration of multimodality. The first tool identified as it is fulfilling these requirements is the MeMo workbench. In its application MeMo finds different interaction paths for a task and identifies potential usability issues. Thereby MeMo employs a probabilistic rule based simulation approach. The other tool identified is CogTool. Using CogTool the designer can predict task execution times of skilled users. Since MeMo predicts interaction path, and CogTool predicts task execution times, both tools differ in their fields of application. By integrating the modality selection algorithm in both tools, the applicability of the mechanism can be tested in these two areas. Furthermore, in a product chain utilizing both tools, certain tasks could be simulated with MeMo in order to gain an understanding about the expected modality usage and the expected interaction steps. Next the same tasks could then be simulated with CogTool to predict the total multimodal task execution time.

2.3 The MeMo Workbench

Practitioners can use the MeMo workbench in early design stages or during iterative evaluations to reveal potential usability issues. Memo was initially used to evaluate models of speech dialog systems (Möller et al., 2006). Jameson et al. (2007) applied the workbench the first time for the evaluation of graphical user interfaces. The simulation of multimodal HCI is not yet possible. MeMo simulates possible interaction between specified user, task, and system models (Engelbrecht et al., 2008). In each step of the MeMo simulation the probabilities of the available interactions are influenced by both characteristics of the user model and the system model, as well as by rules. The user model selects one interaction according to the calculated probability distribution.

The following Sections 2.3.1 to 2.3.5 describe the functionality and the interplay of the most important components of the MeMo workbench. Each section focusses on the description of the functionality of a specific component, whereas the description of the interplay of the components is spread over the single sections. The last section gives fist implications for the integration of an algorithm for modality selection for the simulation of multimodal HCI. Concrete modeling examples with the MeMo workbech are given in Section 5.1.

2.3.1 System Model

The system model is a modeled prototype of a design idea. The MeMo workbench evaluates the system model by simulation of interaction paths. Systems are modeled by making use of a *dialog designer* and a *system designer*. The distinction between

these two components allows for reusability of dialogs containing the same control elements in different system states.

Within the dialog designer MeMo so far supports modeling of graphical dialogs and speech dialogs. Further system models can combine both types of dialogs. However, the simulation of such multimodal system models does not lead to useful results. Once a possibility for speech input is found, the speech path is chosen, otherwise the GUI path. The actual aim of the dialog designer is to create dialogs which are used by the user model to interact with the system. A dialog is therefore a part of the modeled user interface. A graphical dialog includes control elements such as buttons with attributes like label, position, size, etc. Background images can be used in order to integrate screenshots or design sketches of graphical user interfaces. A detailed description of attributes can be found in Schulz (2014). Speech dialogs include spoken system output defining the information required by the system (prompts) and spoken user input defining the information transferred to the system by the user. With regard to speech recognition errors a probability for substitutions, deletions, and insertions can be configured. The exchange of information between the user and the system model is referred to as interaction, whereas input interactions designate the transmission of information from the user to the system model and output interactions designate the transmission of information from the system to the user model. When performing graphical or speech dialogues information is transmitted to the system. Information is represented through attribute-value pairs (AVP). During the simulation information which is annotated within the system dialogs is compared to the user models information about the task (task knowledge) to affect the execution probabilities of interactions.

The actual aim of the system designer is to model the system states and to connect the modeled dialogs. A system state is composed by one or more of the designed dialogs. In order to create the logic of the system transitions between the interactive elements of a state (e.g buttons and speech dialogs) and other composed system states can be defined. Thereby a finite state machine representing the system is built. The execution of a transition can be associated with conditions and consequences. The condition part determines which information currently has to be available in the user knowledge, so that the transition can be executed. The consequence part defines how the available information will be changed after the execution. By means of information changes, the MeMo workbench can determine when a task has been fulfilled by the user model.

2.3.2 Task Model

The task model describes the task to be solved with the system model by the user model. The aim of the task model is to define a start state, the task knowledge and a target state. Further an initial assignment of information can be provided. The task itself is composed of at least one sub-task. Complex tasks can be structured by multiple sub-tasks. Sub-tasks are used to define required information from task

knowledge and corresponding success conditions. Information required in a task is described as attribute-value pairs (AVP). Conditions for successful task completion have to be determined. When all sub-tasks have been completed with success, the main task is also successfully completed.

2.3.3 User Model

The MeMo user model is used during the simulation to interact with the system model. The user model has attributes to describe its characteristics such as age, language skills, physical limitations, psychological attributes, skills, and dynamic user attributes. These attributes are utilized in rules manipulating the probabilities for the selection of input interactions. The structure of the user model is based on the Model Human Processor (MHP) presented by Card (1981). The MHP describes the division of information processing in the three parts of perception, processing and execution. In the Memo workbench, these three parts are represented by interchangeable modules.

The task of the perception module is to perceive information provided by the system and to transfer the perceived information to the subsequent processing. The default perception process is so far implemented as ”complete” perception meaning that all information provided by the system will be fully recognized. Once an option for speech interaction is found, speech is taken, otherwise GUI. A first prototype of a new module for selective perception processing only elements with a particular salience is also available. However, evidence about the extent to which human perception is mapped correctly by the new module is lacking. For the validation of the integration of the extensions for multimodal interaction unknown side effects of immature modules are undesirable. Therefore, the default perception module is used in this work.

The task of the processing module is to decide on the basis of perceived information and the defined task knowledge, which interaction should be applied by the user model on the system model. To accomplish this, the information on the interaction objects transferred from the perception module are compared with the task knowledge of the user model. As a result of this comparison, interaction objects fitting well with the user knowledge are assigned with higher probabilities. After this initial assignment of interaction probabilities the rule engine is called, which further changes the probabilities based on the defined rules.

The basic idea of the rules used in MeMo is to capture typical behaviors of members of specific user groups in specific situations. People with bad eyesight, for example, represent a user group. Rules further take into account the attributes of the interaction object. For a button it can for example be specified that its font size is small. The rules have an ”if then structure” with a condition and a consequence part. In the condition part attributes of user groups and interaction objects are specified, whereas the consequence part defines how the interaction probability of the interaction object will be modified. An example for a rule is therefore: if the user has a

bad vision and the font size of a button is small, then the probability of pressing the button is reduced. Once the rules have affected the probabilities of all available interactions, one interaction object is selected according to the probability distribution and the decision is passed to the execution module. Rules have an XML structure and are stored in a rules folder within the workbench.

The task of the execution module is to apply the chosen interaction on the system model. The execution process is implemented as ”correct”, meaning that the interaction object that was selected, is performed in the right way.

2.3.4 Simulation

Before the simulation can be started, a number of iterations, at least one user group and the tasks to be simulated have to be specified by the modeler. The system model can be represented as a graph consisting of transitions and system states. After the simulation was started at first optimal solution paths through this system graph are calculated to ensure that solutions based on the existing knowledge of the users model exist. Only if solutions are found, the simulation is started. The optimal path is further used to determine during the simulation, if the simulated user leaves this optimal path. The path can then be goal-driven, if the task goal can still be accomplished, or unrecoverable, if the task goal can not be reached anymore. If an interaction ends up in an unrecoverable state, the iteration is aborted, and the next iteration is triggered.

During the simulation the user model interacts with the system model. The following steps are executed:

1. **Initialization:** the user model gets the knowledge for the first sub task; information values and start state of the system model can be set.
2. **Perception:** the user model perceives (all) possible interaction options in the user interface.
3. **Processing:** information is evaluated, rules are applied, and an interaction is selected.
4. **Execution:** the selected interaction is performed.
5. **System reaction:** the system model checks whether a transition can be carried out, the consequences of the transition are applied, and the values of information can be changed.
6. **Task check:** examination if the goal of the sub task is successfully reached.
   - if not, got to 1 with existing initialization.
   - if yes, go to the next sub task and perform the steps with new initialization. If no further sub task exists the whole task is successfully finished.

This process can be carried out iteratively, so that a task can be simulated several times in succession, resulting in log data from many simulated users.

2.3.5 Reporting and Features

In each step of the simulation, the applied rules, the resulting probabilities and the performed interactions are recorded. A report view allows for browsing through the associated data. For each task trial of the user model a path through the system state machine is displayed. The number of needed interaction steps, the correctness of the task solution, and the execution times for single task steps as well as for the whole task can be viewed. Based on this information the developer can understand, why the user model deviated from the optimal path. From that, conclusions on the necessary changes in the design of the user interface can be drawn. A summary of the simulation can be exported as a PDF document for the documentation of the results. For further processing in statistical programs a more detailed log file can be output in CSV format.

2.3.6 Implication

With MeMo practitioners can create models of user interfaces and of tasks to be solved by means of these interfaces. Annotated interface attributes, and the AVPs modeling information as well as the task knowledge are effective factors influencing the simulation. With MeMo multimodal systems can be modeled, but human behavior when selecting input modalities in multimodal interaction can not be simulated correctly because a corresponding mechanism is missing. MeMo choses the speech path through the system graph once a possibility for speech input is found. Only if no possibility for speech input is found a GUI path is selected. This principle preference for speech input in case of multimodal system models must be overcome. The module-based approach of the MeMo workbench and in particular the subdivision of the user model into several modules allow for the exchange and change of individual modules. An available algorithm for modality selection could thereby be integrated. In order to enable the integration, it must be ensured that input data of the algorithm are available when needed. The input data in this case include concrete values for input performance and modality efficiency of the different modalities. The input performance of speech input is already considered within MeMo and should be easy to exploit. However, for other modalities error rates are not foreseen. Modality efficiency as described in Section 2.1.1.3 is measured in terms of interaction steps. This parameter is so far not used during the simulation. In each interaction step it is necessary to enable the user model to get information about the number of interaction steps to be expected for each modality. This information could be derived if the current state and the end state of the current sub goal are known and if the single path steps on the optimal path between these states can be calculated in advance. The extensions made to MeMo, as we'll as application examples showing the power and the effort of the approach can be found in Section 5.1. In the next section CogTool is introduced.

2.4 CogTool

The details shown in this section are mostly taken from the CogTool user guide (John, 2012). CogTool is a user interface prototyping tool that can produce quantitative predictions of how users will behave when the prototype is ultimately implemented. Thus, CogTool provides a way to explore different UI ideas, compare them, and narrow down the options to a handful of designs to be empirically tested with users. Using CogTool the task execution time for skilled users of a system can reliably be predicted (John, et. al., 2004; Luo & John, B., 2005). CogTool employs the ACT-R cognitive architecture. It should be noted that the concepts underlying CogTool are partly similar to those already described in the previous section about MeMo. However, the naming is different in CogTool. In this section the CogTool naming is used.

The Sections 2.4.1 to 2.4.5 describe how to create the necessary models, and sketches how the predictions of underlying human performance model are made and how they are reported. Concrete modeling examples with CogTool are given in Section 5.2.

2.4.1 Prototyping an Interface

The prototyping of an interface is explained in detail in the CogTool user guide (John, 2012). Here only the key elements of the underlying prototyping concept are described in adapted excerpts of the user guide.

In CogTool a prototype of a system is represented as a ”design”. The design represents a finite state machine that consists of frames and transitions between those frames. The frames contain devices for user input. Starting from a frame a transition defines the move to another frame if a particular input device is used. A device is a representation of the hardware associated with the design. CogTool includes the input devices keyboard, mouse, touchscreen and microphone, as well as the output devices display and speaker. CogTool assumes that every design has a display. For the graphical input devices several widgets, such as buttons, check boxes, hierarchical menus, etc. exist. It is not possible to deselect the display as an output device. If a device with no display should be modeled (e.g., a speech dialog system), this can be handled by all frames being empty. Speech input from a user is represented by a microphone that can be included in the design. When modeling a transition for speech input, the words for this particular utterance have to be defined.

Using CogTool to make predictions of task execution time for skilled users, it is not necessary to integrate an input option for every interactive element in the design. The underlying human performance model only needs the input options that are actually used in the tasks under investigation. With regard to multimodal HCI, multimodal designs are in principle possible. Between two frames one transition from a graphical widget and another transition from a microphone can exist.

2.4.2 Task Demonstration

The demonstration of tasks is explained in detail in the CogTool user guide (John, 2012). Here the key aspects of the task demonstration are described in adapted excerpts of the user guide.

As soon as a design including the frames and transitions that are necessary for conducting a specific task is created the exact interaction steps of this task have to be demonstrated in the UI mockup. Before the actual task demonstration can begin a start frame has to be selected. During the presentation the modeled input options of the graphical user interface and other input devices like the microphone are used. The modeler can interact with the frames in a way similar how a user would interact with the actual device. Multimodal task demonstrations are also possible. If for example one transition from a graphical widget and another transition from a microphone exist and the task can be solved via both transitions the modeler can pick one of these transitions. In CogTool each selection of a transition is performed by the modeler, and if different input devices within one state include transitions to the same subsequent state the modeler also has to select the desired device.

CogTool automatically adds steps to the demonstration in order to create cognitively plausible scripts. Most of these steps are ”think” steps, placed in accordance to prior research that has studied where people pause when using computers (e.g., Card, Moran, and Newell, 1980; Lane, et. al., 1993). The rules for placing ”think” steps can be looked up in the CogTool user guide. It is further possible to remove or edit the duration of ”think” steps placed by CogTool and to add additional ”think” steps with a configurable duration to the demonstration. In the user guide it is discouraged to edit or add ”think” steps unless one has empirical evidence applicable to the design to support the change. In order to get information from non-interactive parts of the graphical user interface additional steps to ”look at” particular widgets can be added. This can for example be used to read a dialog box in the design.

2.4.3 Human Performance Model

CogTool’s quantitative predictions are based on the cognitive architecture ACT-R (Anderson and Lebiere,1998). ACT-R is used to simulate the cognitive, perceptual and motor behavior of humans interacting with the prototype to accomplish tasks the UI designer has defined (John, 2012). In ACT-R knowledge about how to perform a specific interaction is represented by productions. CogTool converts the recorded interaction steps into a sequence of such ACT-R productions. The generated code is executed in the ACT-R Runtime Environment. The simulation calculates the time of an expert user for the execution of the steps. The model that CogTool creates is based on the Keystroke-Level Model (KLM; Card et al., 1980). CogTool’s predictions of human performance use Fitts’s Law to estimate movement time but since Fits’s Law was originally determined using tapping with a stylus, there is no additional time added for the touch screen input. CogTool’s predictions for speech input use ACT-

R's speaking model. That model uses 50 ms per character as an estimate for how long it takes the user to say words into the microphone. While demonstrating the task CogTool automatically inserts mental operators ("think" steps) into the KLM.

2.4.4 Computing a Prediction

The computing of predictions is explained in detail in the CogTool user guide (John, 2012). Here the key aspects of the computing are described in adapted excerpts of the user guide.

CogTool transforms the demonstrated task into a script, which is translated into cognitively valid code of the cognitive architecture ACT-R. This code is run in ACT-R and produces the prediction of execution time. Cognitively valid means for example that validated models for movements of fingers on a touch screen, and a speaking model as well as a hearing model are being utilized to compute the prediction of performance. Once creating and editing a script is finished, the computing of a prediction works straightforward by pressing button. The result is the calculated prediction for the execution time of a skilled user.

2.4.5 Reporting

In the visualization window of CogTool details of the cognitive simulation can be viewed and different designs as well as tasks can be compared to each other on a timeline. The activities on the single ACT-R modules and the simulation traces can be viewed. Based on this information the practitioner gains insights about the efficiency of the designs. It is easy to recognize how long each interaction step takes, and where improvements of the design may be made. In this way one can also decide for a specific design. For documentation the simulation results (expert time predictions) of multiple designs and the script generated by CogTool can be exported in CSV format. Further the generated ACT-R model file can be exported as lisp programming code. This code can also be run directly in ACT-R. Also including information about the involved interaction steps, the trace can be exported in a text document. One can also export prototypes to HTML, to share designs with colleagues or to perform quick user tests.

2.4.6 Implications

With CogTool practitioners can create designs of user interfaces and demonstrate tasks using the designs. The results of the predictions are execution times of skilled users for the tasks demonstrated on the UI mockup.

As depicted above multimodal designs and task demonstrations are possible with CogTool. CogTool can compute performance predictions with mixed modality usage. If differences between tasks including divers sequential combinations of input modalities are to be explored, alternative task demonstrations have to be recorded for each available modality combination of interest. A mechanism automatically generating these combinations would allow a comprehensive examination of different task solutions, and save modeling effort and time.

While demonstrating a task in CogTool, frames including a touch screen and a speech transition, both allowing to solve the task, could for example integrate a ”select modality” step (similar to the ”think” or ”look at” steps) which could be part of the CogTool script. After the ”select modality” step the modeler would demonstrate both the touch screen and the speech solution, resulting in two CogTool sub scripts, which are recombined in a later frame (at the latest in the last frame). In this way the input options for multimodal interaction could be transferred to the underlying human performance model basing its calculations on the user inputs that are actually performed in the tasks under investigation. The modality selection mechanism then has to be integrated on the part of the human performance model.

In the core, CogTool uses ACT-R as a human performance model to perform simulations. In this respect one application of the CogTool functionalities is, that it can be used as graphical user interface for generating ACT-R simulations. An extension of CogTool by functions for examining modality selection in multimodal HCI, thus involves adaptations on both the CogTool UI part and the ACT-R simulation part. Planning an integration of modality selection into CogTool, it makes sense to firstly test the feasibility and validity of a multimodal ACT-R simulation. The multimodal ACT-R simulation should be orientated on the structure of the ACT-R code that is generated from CogTool scripts. Further correctness of input data of the modality selection algorithm has to be ensured. The procedure to simulate multimodal HCI based on CogTool generated ACT-R models, as well as application examples showing the power and the effort of the approach can be found in Section 5.2. In the next section the research questions of this work are formulated.

2.5 Research Questions

The motivation of this work is to examine the interdependencies of the factors modality efficiency and input performance that determine modality choice, for the domain of multimodal mobile HCI, and to build models that are able to predict modality usage by means of these factors. The selected task presented in Chapter 3 is from the domain of ”list browsing”, which is of importance for a number of systems, especially for smart phone applications where screen size is limited. Since not all information can be displayed at a time, users have to browse through lists to find desired items.

Some questions that may arise in certain contexts of this work will not be answered, as they exceed the scope of this work. For the simulation of errors for ex-

ample no additional implementations are made in order to limit the scope of this work. With MeMo ASR errors but not errors of the graphical user interface can be simulated, and with CogTool no errors of input devices can be simulated at all. The simulation of errors can be complicated, since different errors may result in different system states. The realistic simulation of errors is therefore an own field of research.

2.5.1 Quantification of Modality Efficiency and Input Performance

The decision mechanisms of human beings regarding modality selection are so far not fully understood. The available theories and research findings about factors influencing modality selection provide starting points for experimental studies. In order to derive computational models of modality selection comparable data has to be gathered in unified investigations. This points out to the first research gap that is studied in this work: the lack of empirical findings regarding modality selection. Therefore the following research question RQ1 is formulated:

- **RQ1**: Can significant effects of modality efficiency and input performance on the selection of input modalities in multimodal HCI be disclosed by unified experimental investigations?

According to the existing theories for the choice of modalities experiments investigating relevant influence factors have to be designed in a way, that the necessary information for modality selection is available for the participants. Accordingly strategies of human decision-making like heuristics have to be considered in the design of experiments. In order to answer RQ1 three interaction experiments with a SIMS are presented in Chapter 3, testing the following hypotheses regarding modality selection:

- **H1**: If the input performance of a specific modality decreases, the usage of this modality decreases as well.
- **H2**: If the modality efficiency of the touch screen interface decreases, the usage of speech input increases.

2.5.2 Computability of Modality Selection

Computational models for modality selection are so far missing completely. Therefore the derivation of such a model for modality selection is the aim of the second research gap. The state of the art in AUE and the developments in AUE of multimodal systems presented in this chapter implicate that cost benefit approaches employing utility functions could be applied for model creation. Therefore the second research question RQ2 is formulated as follows:

- **RQ2**: Can a utility-driven computational model of modality selection be formed based on empirical data?

In order to answer RQ2, in Chapter 4 models are derived and the predictive quality of the tested models is judged with goodness of fit measures (R^2 and $RMSE$), as well as with data interpolation and extrapolation. To show the general applicability for simulation, a prototypical implementation of the prediction algorithm will be outlined.

2.5.3 Application for Automated Usability Evaluation

AUE tools integrating automated modality selection are so far missing. The possible application of the built model depends on the compatibility with existing simulation tools. If automated usability evaluation of multimodal systems should be enabled ways to integrate the model have to be found. The third research question RQ3 is therefore:

- **RQ3**: Can the compiled model for modality selection be utilized for the practical application in the field of automated usability evaluation?

RQ3 will be answered in Chapter 5 by the application of the modality selection algorithm within the two AUE tools MeMo and CogTool. Regarding MeMo a full integration of the model could be performed. For CogTool a multimodal procedure for simulating modality selection based on adapted ACT-R models generated by CogTool is developed.

2.6 Chapter Summary

In this chapter basic principles and recent developments of multimodal human computer interaction have been described. The selection of input modalities in multimodal HCI involves a decision process of the user. Therefore also basic principles of human decision-making have been discussed. It was argued that existing theories of multimodal information processing and human strategies for decision-making have to be considered during the planning of experiments investigating modality selection. In order to support the creation of a utility-driven model the influencing factors of modality selection assessed during such experiments should be related to efficiency and effectiveness. The factors input performance and modality efficiency were identified as suitable. According to recent developments in the filed of multimodal AUE the tools MeMo and CogTool have been identified as candidates for testing the application of the modality selection algorithm to be developed. For both tools first starting points for the utilization of the modality selection algorithm were identified. Lastly the research questions of this work were formulated.

In the next chapter three interaction experiments assessing modality efficiency and input performance in several levels are presented.

Chapter 3
Effects of Modality Efficiency and Input Performance on Modality Selection

In this chapter a series of three experiments will be described. The overall goal of all experiments is to gather comparable data on users' modality choice behavior. Therefore all participants have to conduct the same tasks with one particular system in controlled laboratory setups. Modality efficiency is systematically varied in six levels within each experiment. Throughout the experiments six conditions of input performance are differentiated.

Section 3.1 describes the general experimental setup for all experiments. The first experiment depicted in Section 3.2 aims at the mere effect of modality efficiency. It comprises the baseline condition for input performance [T00, S00] with touch screen T and speech input S having both no errors. Perfect input performance for both modalities was simulated using Wizard of Oz speech recognition (Schaffer and Reitter, 2012). In Section 3.3 the second experiment is described, targeting the effect of input performance, ASR error rates of 10% ([T00, S10]) and 30% ([T00, S30]) were simulated (Schaffer et al., 2011a). In the third experiment depicted in Section 3.4, ASR errors of 20% ([T00, S20]), as well as two further conditions were tested: [T20, S00] touch input producing errors at a rate of 20% while ASR errors amounted to 0%, and [T20, S20] with touch and speech input, both comprising error rates of 20% (Schaffer and Minge, 2012).

3.1 Experimental Setup

All experiments followed roughly the same experimental setup. Specifically the used system, the task and the general procedure were extensively harmonized. Adaptions to the general procedure and the simulation of input performance are explained within the respective subsections of the single experiments. Further details of the experiments can be looked up in Appendix A.

3.1.1 Interactive System

3.1.1.1 Application

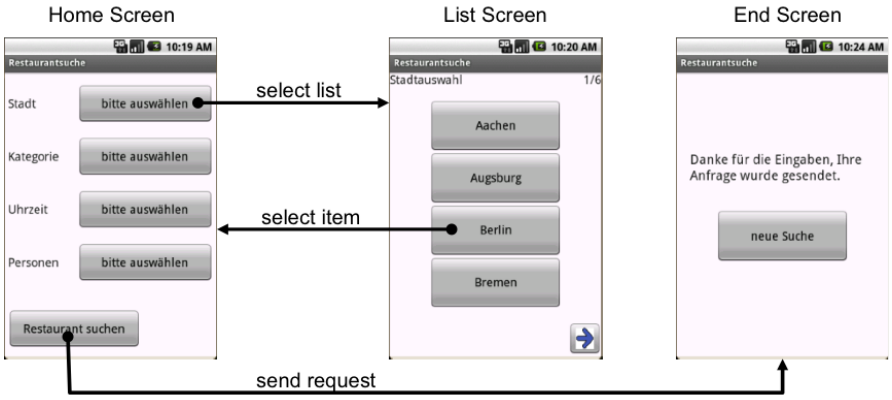

Fig. 3.1 Screens of the 'Restaurant Booking Application' (RBA).

For all experiments, a prototypical smartphone-based app called the 'Restaurant Booking Application' (RBA) was used. The app ran on an Android device (G1 HTC) and any possible interaction could be made using touch screen, speech input, or both sequentially. Speech input was directly possible at each point in the interaction. There were four slots that had to be filled by the user to accomplish the task of booking a restaurant: city, cuisine, desired time, and number of people. For each slot, the user had to choose his request from a list. Each list contained 24 items and was split into 6 layers, each containing 4 items. To make a request, the user had to press a category button on the touch screen or speak out the written label and then step through the layers to find the designated item (see Figure 3.1). After making a selection by touching an item or verbalizing it, the user was redirected to the home screen. When all slots were filled, the user was able to send his request, accomplishing the task, and the end screen was shown.

To simulate ASR, a Wizard-of-Oz design was used. Therefore an open microphone was hidden in the room the participant was working in. In a different room, unseen by the participant was the 'wizard', a human operator wearing headphones. The wizard listened to the participants' speech input and used a graphical user interface (a Java application on a Linux notebook), to execute the voice commands. Following the WIMP (window, icons, menu, pointer) concept, the wizard interface was designed in a way that responses could quickly be generated. The wizard tried to finish the input about 0.5-1.0 seconds after an utterance. The feedback was sent to the mobile device via TCP-IP and wireless LAN, simulating the ASR system. The simulation of errors is separately described for each experiment. The participants

assumed that ASR worked with an open microphone built in the smartphone device. It was not necessary to push a button to enter a speech command. The language was German for both the touch screen and speech-based interface.

3.1.1.2 Benefit of Speech Usage

To make a request using the touch input, the user had to click on a category and then manually switch through layers to find the desired item. In contrast to this, speech interaction offered a shortcut. Using speech, the user could not only make exactly the same inputs as with the touch screen but also choose an item from any list screen without the need to switch through layers by simply pronouncing the name of the desired item. For example, one was able to select 'Hamburg' from the first layer, while it would have taken him four interaction steps to select it using the touch screen. The benefit of speech usage B_{speech} thereby can be calculated as the difference between necessary touch screen interaction steps IS_{touch} and speech interaction steps IS_{speech}. [1]

$$B_{speech} = IS_{touch} - IS_{speech} \tag{3.1}$$

3.1.1.3 Consideration of Heuristics

The participants have to conduct three rehearsal trials: one using touch screen input only, one using speech input only, and one where modalities can be selected as desired in each interaction step. While performing the tasks the participants experience the presence or absence of errors dependent on the condition of input performance. The rehearsal trials are structured in such a way that different levels of modality efficiency can be experienced. The error rates of input modalities and the modality efficiency may have effect on the perceived usefulness of the modality. It is expected that the participants perform an internal cost-benefit estimation in order to weight up the usefulness of modalities. In this way an anchoring with respect to the usefulness of the modalities is made.

By performing the rehearsal trials also the availability of a modality may be affected, as the users gain initial experience with the use of the input modalities. Especially for participants using speech input for the first time it may become easier to imagine the use of this modality during the experiments. Negative experiences like

[1] For the RBA IS_{speech} amounts to a constant 1, as speech input is always directly possible. However IS_{speech} can amount to other values in different systems. Alternatively, using an OKS setup for the actual system (compare Section 1.1), IS_{speech} could amount to a constant 2, due to a necessary ASR activation command. If no speech shortcuts were implemented in the actual system and a speech command would still be implemented for each possible touch input IS_{speech} would equal to IS_{touch}. For other systems (like some navigation systems), where speech input does not directly match touch input, varying values of IS_{speech} are possible. As our aim is to create a model that can in principle cover all these kinds of systems, IS_{speech} will be handled as variable in the model construction.

ASR or touch screen errors may also cause that an error-prone modality is used less frequently.

A cue about modality efficiency can also be derived from the graphical user interface: within the list screens the actual number of a sub list and the total number of sub lists is shown in the upper right corner (comp. Figure 3.1.1.1). As the list items are ordered alphanumerically the number of touch screen inputs to perform is easy to estimate, and easy to compare to speech input. For users experiencing speech shortcuts the use of speech may appear particularly representative for tasks with high modality efficiency of speech input.

3.1.2 Task

In all experiments the task was to perform predefined database requests using the RBA. Each request comprised four list-browsing sub tasks, namely the choice of a city, a culinary category, a time, and a number of persons. For example, participants had to request for "a Chinese restaurant in Berlin at 8 pm for 12 persons". The exact steps and system states for this task example are described in Table 3.1. At first a list had to be selected at the home screen. In doing so a user can choose touch screen or speech input at any time. The participants were instructed that the written labels in the GUI have to be used as speech commands. After the selection of a list the first layer was presented. The 24 items within a list are ordered alphanumerically on 6 layers, whereas each layer presented 4 list items in the GUI. Figure 3.1 depicts the choice of the city 'Berlin' as an example. The participants were instructed that touch screen or speech input could be used to select an item. Using speech input all list items from all six layers of a list are already recognized in the first layer, while touch screen input only allows selecting items that are directly visible. Users preferring touch screen input therefore have to browse through the list by pressing the arrow in the downright corner, until the desired list item is displayed. Due to the alphanumeric order, the participants should be able to anticipate the layer of a distinct item.

3.1.3 General Procedure

All experiments followed roughly the same procedure. Adaptions can be found in the individual subsections. However the general procedure depicted here was uniform for all experiments. After welcoming the participants, demographic data was gathered in a short questionnaire. Then, the participants were given an introduction, explaining the usage of the system with both the touch screen and speech input. To control whether participants understood the instruction and were able to use the system with both modalities, three training trials had to be done: one using touch input only, one using speech input only, and one with a free choice of modality.

Table 3.1 Task example for 'Look for a Chinese restaurant in Berlin at 8 pm for 12 persons'. From top to bottom the column System state indicates the states involved during task processing. Sub goals are marked with SG. B_{speech} is calculated as described in Subsection 2.1.2. The columns 'touch button' and 'speech utterance' explain the user input interaction.

System state	B_{speech}	touch button	IS_{touch}	speech utterance	IS_{speech}
Home	0	Press "select City"	1	Say "select city"	1
City list 1	0	Press "Berlin"	1	Say "Berlin"	1
Home (SG1)	0	Press "select Category"	1	Say "select category"	1
Category list 1	1	Press "right arrow"	2	Say "Chinese"	1
Category list 2	0	Press "Chinese"	1	- " -	1
Home (SG2)	0	Press "select Persons"	1	Say "select persons"	1
Persons list 1	3	Press "right arrow"	4	Say "twelve"	1
Persons list 2	2	Press "right arrow"	3	- " -	1
Persons list 3	3	Press "right arrow"	2	- " -	1
Persons list 4	0	Press "12 persons"	1	- " -	1
Home (SG3)	0	Press "select time"	1	Say "select time"	1
Time list 1	2	Press "right arrow"	3	Say "eight"	1
Time list 2	1	Press "right arrow"	2	- " -	1
Time list 3	0	Press "8 pm"	1	- " -	1
Home (SG4)	0	Press "Search Restaurant"	1	Say "search restaurant"	1

After this training, the target trials started. Adaptions regarding the simulation of errors are described in the individual subsections. Participants were informed about the Wizard-of-Oz design only after all target trials were finished. Depending on the specific conditions of the experiments, input performance could be impaired. However the participants were not informed about the possible occurrence of errors beforehand. As the measure of modality efficiency, B_{speech} was systematically varied between 6 levels (0-5 interaction steps) within the individual trials. Modality usage data was collected using log files. This experimental setup enabled to measure the effects of modality efficiency and input performance on modality usage by means of two independent variance analyses. During the experiment, also data about the perceived mental effort and product quality was gathered. This data will not be further examined here (see Schaffer and Reitter (2012), Schaffer et al. (2011a) and Schaffer and Minge (2012) for more information). All experiments took roughly 45 minutes to one hour and were remunerated with € 10.

3.2 The Influence of Modality Efficiency

This experiment investigates the influence of modality efficiency on modality selection. Shortcuts of the speech interface lead to a higher modality efficiency of speech compared to touch screen. Here only results regarding modality selection are reported. Further results can be looked up in Schaffer and Reitter (2012).

3.2.1 Adapted Procedure

In the first experiment 16 German-speaking participants (8 females) tested the RBA. The age ranged from 22 to 31 years ($M = 26$, $SD = 2.95$). The course of the experiment corresponded to the general procedure described in Section 3.1.3. The condition [T00, S00] was tested. Since speech input and touch input were both working perfectly, neither producing any errors. For each of the 15 trials (including 3 training trials), an instructor was presenting the current task on a paper.

3.2.2 Results

Tests on distribution form and the homogeneity of variance can be looked up in Appendix A.15. The means and standard deviations of speech usage are reported in Table 3.2. A one-factorial repeated measures ANOVA showed a highly significant effect of B_{speech} on the usage of speech ($F(2.27, 33.97) = 27.503$; $p_{1-tailed} < .001$; $\eta_p^2 = .647$). The results confirm the hypothesis H2 that modality efficiency in terms of benefit with reference to interaction steps is moderating users' modality choice. Speech usage increases with increasing efficiency of the speech modality.

Table 3.2 Experiment 1. Means M and standard deviations SD of speech usage in dependence of B_{speech}.

Condition	B_{speech}	0	1	2	3	4	5
[T00, S00]	M [%]	0.31	0.73	0.83	0.90	0.94	0.96
	SD [%]	0.36	0.26	0.22	0.18	0.09	0.08

3.2.3 Discussion

The results confirm the hypothesis H2 that modality selection is influenced by the efficiency of the modality. If desired items are to be found in a deep-set layer of a list, speech usage is preferred over touch input due to shortcuts only available by speech. Apart from that users tend to use touch if the benefit of speech equals zero. The preference of touch input in this case indicates that other influencing factors come into play if only one interaction step is necessary for both modalities. As speech still can be considered as a novel input modality the familiarity of touch interaction might be one important factor for modality selection if speech does not

offer a shortcut. However the interaction behavior could change if speech input is established in more interfaces and thus gets more familiar.

Another explanation could be that touch input was more efficient in terms of time. The average duration of one interaction step was longer for speech input. Further cognitive workload of speech processing compared to touch usage is assumed to be higher (McCracken and Aldrich, 1984). The factors efficiency in terms of time and mental effort are known influence factors of modality selection. Effects of modality efficiency and input performance on mental effort are reported in Schaffer et al. (2011b), Schaffer et al. (2011a), and Schaffer and Minge (2012). Future research will be needed in oder to consider interaction time related measures and mental effort in computational models for modality selection.

3.3 The Influence of Input Performance

This experiment investigates the influence of speech input performance of on modality selection. Modality efficiency is varied in the same way as in experiment 1. Here only results regarding modality selection are reported. Further results can be looked up in Schaffer et al. (2011a).

3.3.1 Adapted Procedure

In the second experiment, 33 German-speaking participants were tested. Four participants had to be excluded as three did not follow the instructions and one experienced a severe malfunction of the system. The remaining sample consisted of 11 females and 18 males, with a mean age of 25 years ($SD = 3.7$). The course of the experiment corresponded to the general procedure described in Section 3.1.3. Error rates were generated randomly, varying between 0 and 40 percent. Thus the error rate was individual and varied between participants. The participants were clustered in two groups post hoc. The first group (constituting the condition [T00, S10]), consisted of 17 subjects who experienced error rates of approximately 10% ($er = 0-20\%$, $M = 9.13\%$, $SD = 6.0\%$). The second group (constituting the condition [T00, S30]) contained 12 subjects who experienced error rates of approximately 30% ($er = 21-40\%$, $M = 30.23\%$, $SD = 4.28\%$). ASR errors were randomly inserted and could occur in each step. In terms of the error probability corresponding to the individual error rate, it was possible that errors were induced directly in succession. After an error the home screen was presented with no result and the textual message 'I did not understand', and the search within the list screen had to be started again. The 3 training trials and 15 target trials per user took place in the same experimental setup as in the first experiment.

3.3.2 Results

Tests on distribution form and the homogeneity of variance can be looked up in Appendix A.15. The means and standard deviations of speech usage are reported in Table 3.3. Corresponding with the first experiment a one factorial repeated measures ANOVA showed a highly significant effect of B_{speech} on the usage of speech ($F(2.47, 65.79) = 74.222$; $p_{1-tailed} < .001$; $\eta_p^2 = .733$). Further significant differences between the two error conditions could be observed ($F(1,27) = 6.94$; $p < .007$; $\eta_p^2 = .204$). The results confirmed the hypothesis H1 that input performance moderates users' modality choice. Speech usage decreases with increasing ASR error rate. Further H2 is confirmed again. A significant interaction between both factors could not be observed.

Table 3.3 Experiment 2. Means M and standard deviations SD of speech usage in dependence of B_{speech}.

Condition	B_{speech}	0	1	2	3	4	5
$[T00, S10]$	M [%]	0.35	0.75	0.92	0.93	0.90	0.93
	SD [%]	0.32	0.30	0.12	0.16	0.17	0.19
$[T00, S30]$	M [%]	0.17	0.66	0.70	0.81	0.77	0.84
	SD [%]	0.18	0.26	0.25	0.11	0.22	0.18

3.3.3 Discussion

The results confirm the findings of experiment 1 indicating that modality selection is influenced by modality efficiency. If desired items are to be found in a deep-set layer of a list, speech is preferred over touch screen due to shortcuts of speech input.

The results of experiment 2 further show that speech as an input modality is less used if the probability of ASR errors increases. Users adapt their interaction behavior to the accuracy of a system when they decide which modality they use next. As the accuracy of future ASR modules might increase the influence of this factor might decrease or even fade away. But interaction designers of presently developed systems have to be aware of the reliability of their input modules.

If a system is affected by ASR errors on the one hand and contains speech shortcuts on the other hand, a user has to distinguish which modality is more capable. The threshold at which speech becomes more efficient shifts with increasing error rate and the advantage of speech shortcuts even may completely vanish if the error rate is too high.

The study further reveales that users seem to be aware of the changing characteristics of certain factors and try to weigh up the best decision.

3.4 Combined Effects of Input Performance of Touch Screen and Speech

This experiment investigates combined effects of touch and speech input performance on modality selection. Modality efficiency is varied in the same way as in the experiments 1 and 2. Here only results regarding modality selection are reported. Further results can be looked up in Schaffer and Minge (2012).

3.4.1 Adapted Procedure

Finally, in the third experiment, 48 German-speaking subjects (24 m, 24 f) participated. The mean age was 24.2 years ($SD = 3.73$). The course of the experiment corresponded to the general procedure described in Section 3.1.3. Unlike in experiment 2 (where ASR errors were simulated randomly) ASR errors were simulated with fixed error rates of 0% or 20%. Touch screen errors were integrated by blocking one in five input attempts on the touch screen for 1.4 seconds, the average time needed to recover after an ASR error. The blocking mechanism was implemented so that it counted across individual tasks. Therefore touch screen errors could also occur in the first step when using touch input as off the second task. Due to speech input involvement, touch screen errors were further distributed over the interaction steps. At the end of the experiment, each subject was asked about the perceived authenticity of errors. None of the participants noticed that errors were simulated, and none stated that they had recognized the errors as artificial. The participants were evenly distributed to the following four error conditions:

1. [T00, S00] touch and speech input both working perfectly (as in the first experiment)
2. [T20, S00] 20% touch screen error rate and speech input working perfectly
3. [T00, S20] touch input working perfectly and 20% ASR error rate
4. [T20, S20] 20% touch screen error rate and 20% ASR error rate

A total of 3 training trials and 12 target trials were conducted in a closed acoustic booth. The instructor was sitting outside and the subject was alone during the trials. The tasks were automatically presented on a screen when the previous task was accomplished.

3.4.2 Results

Tests on distribution form and the homogeneity of variance can be looked up in Appendix A.15. The means and standard deviations of speech usage are reported in Table 3.4. Consistent with the first and the second experiment, a one factorial repeated measures ANOVA showed a highly significant effect of B_{speech} on the usage of speech ($F(2.96, 130.32) = 16.72$, $p < .001$; $\eta_p^2 = .275$). Further a highly significant effect of the error condition on speech usage could be observed ($F(3,44) = 4.75$, $p = .006$, $\eta_p^2 = .25$). Post-Hoc Scheffé tests showed significant differences between the conditions [T00, S00] and [T20, S00], as well as between [T00, S00] and [T20, S20]. The results confirmed that previously unconsidered touch screen errors also affect modality choice. Speech usage increases if touch screen errors occur. The study further revealed that touch screen errors are punished harder than ASR errors, as only marginal differences between the [T20, S00] and [T20, S20] conditions could be observed for high efficiency of speech input. A significant interaction between benefit and error rate could not be observed.

Table 3.4 Experiment 3. Means M and standard deviations SD of speech usage in dependence of B_{speech}.

Condition	B_{speech}	0	1	2	3	4	5
$[T00, S00]$	M [%]	0.40	0.55	0.68	0.74	0.79	0.69
	SD [%]	0.36	0.39	0.36	0.31	0.30	0.33
$[T20, S00]$	M [%]	0.75	0.89	0.88	0.87	0.86	0.89
	SD [%]	0.35	0.21	0.21	0.25	0.25	0.23
$[T00, S20]$	M [%]	0.37	0.72	0.71	0.80	0.73	0.86
	SD [%]	0.32	0.28	0.24	0.28	0.32	0.29
$[T20, S20]$	M [%]	0.59	0.86	0.93	0.92	0.91	0.88
	SD [%]	0.26	0.18	0.08	0.22	0.11	0.21

3.4.3 Discussion

The results confirm the findings of experiments 1 and 2 indicating that modality selection is influenced by modality efficiency. If desired items are to be found in a deep-set layer of a list, speech is preferred over touch screen due to shortcuts of speech input.

The results of the condition [T00, S20] with a 20 % ASR error rate are in line with the findings of experiment 2 where ASR error rates of 10% and 30% were simulated. The condition [T00, S00] with no error-prone modalities, was expected to have a higher difference compared to the [T00, S20] condition. In experiment 2

higher speech usage was clearly visible if less ASR errors occurred. The results of a significance test comparing the baseline conditions [T00, S00] of experiment 1 and 2 performed in the next section however reveals no significant differences. Regarding both conditions with touch screen errors the results show significantly increasing speech usage. Once touch screen input fails speech usage increases regardless of whether ASR errors arise or not. Referring to modality usage in the condition [T20, S20] it is concluded that touch screen errors are punished stronger than ASR errors. One reason for this result could be that in today's interfaces the probability for touch screen errors is usually relatively low.

Overall the study reveals that users seem to be mostly aware of modality efficiency and input performance and adapt their modality usage to these factors. The results confirm that merging touch screen and speech input into multimodal interfaces is desirable for domains like list browsing as users can make use of more efficient modalities.

3.5 Resulting Database

The series of experiments was designed for minimizing inter-experimental differences. However comparability may be limited to some extent, as changes in laboratory rooms, task presentation method, and user groups appeared. As baseline condition data comprising errorless interaction for both input modalities was gathered in experiments 1 and 3. No significant differences in modality choice could be found between the respective data sets ($F(1,26) = 2.54$; $p < .123$; $\eta_p^2 = .089$). In order to ensure a better comparability of the individual conditions with respect to the size of the sample only baseline data from experiment 1 was used. Taken together data from six different error conditions comprising 82 participants was gathered and merged into one database.

The database was checked for outliers potentially causing significant degradation of a model's predictive power. Two criteria have been considered. The first pertains to the slope of the linear model computed from the modality usage profile of individual participants. With respect to the effect of the levels of interaction steps, the slope should be positive for all participants of all experiments. If the slope is negative the participants' modality choice trend is contrary to average user behavior. Considering all experiments, the slope of 5 participants turned out to be negative. It was determined that participants with a slope smaller than -0.5 should not be used for model development, as their data will significantly reduce model performance. One participant from the [T00, S20] condition of experiment 3 did not match this requirement and was therefore omitted. Furthermore, the impairment of model performance is less intense, if negative slopes are small and the average amount of speech usage is relatively high. Thus, as a second criterion, it was determined that participants should be excluded if the slope was negative (between -0.5 and 0) and average speech usage was smaller than 50%. One participant from the [T20, S00] condition of experiment 3 was therefore omitted from the model development. Dif-

ferences between Figure 3.2 and the tables reporting the results of the experiments can be attributed to the omission of outliers.

For each of the 80 remaining participants the resulting database contains the error condition and the average speech usage for each level of benefit. Generated by averaging speech usage over all participants of a condition, the speech usage curves in Figure 3.2 yield a summary of the data. All curves are arranged according to the error conditions. All graphs illustrate the percentage of speech usage (y-axis) as a function of B_{speech} (x-axis). Despite certain inter-experimental differences the curves are mostly consistent with each other. Regarding all ASR error conditions, speech usage increases with increasing B_{speech}. Furthermore speech usage decreases with increasing ASR error rate. Only minor differences can be observed between error curves of 0% and 10%, as well as between 20% and 30%. The touch screen error conditions, consisting of a baseline and a 20% touch screen error curve, show mostly consistent behavior. Speech usage increases in the presence of touch screen errors. Regarding the 20/20% curve in the mixed condition speech usage is decreased for $B_{speech} = 0$, as the speech input cannot assert its benefit in efficiency. For higher levels of B_{speech}, speech usage increases and the curve runs similar to the 20% touch screen error condition curve. As both ASR and touch screen errors are present in the mixed condition, this implies that touch screen errors are punished more severely then ASR errors.

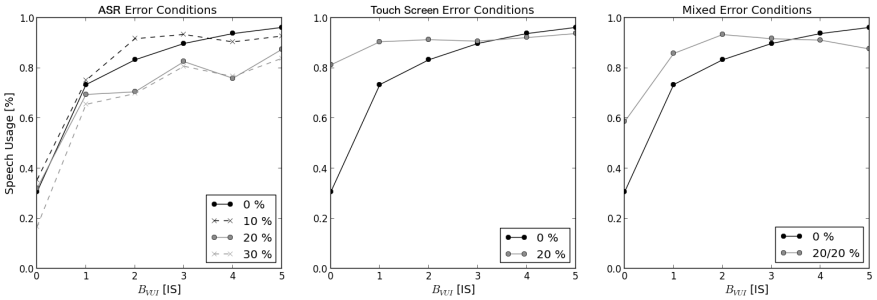

Fig. 3.2 Modality usage curves gathered from three experiments. The 0% baseline was gathered in experiment 1. The 10% and 30% ASR error conditions were gathered in experiment 2. The 20% ASR error condition as well as the 20% touch screen error condition and the 20/20% mixed error condition were gathered in experiment 3.

3.6 Chapter Summary

In this chapter a series of three experiments with a prototypical multimodal system called the restaurant booking application (RBA) were described. The RBA integrates touch screen an speech input. During the design of the RBA and in the design

of the RBA experiments it was attempted to consider strategies of human decision-making. Looking for restaurant comprises list browsing tasks, where speech usage is more efficient, if the desired item is to be found in a deep-set layer of a list. During the experiments modality efficiency of the speech interface and input performance of both modalities were independent variables. The results of the experiments reveal mostly consistent modality selection behavior. On the one hand the usage of speech increases with an increasing modality efficiency of speech (H2). On the other hand the usage of a modality decreases if its input performance decreases as well (H1). The results of all experiments were merged into one database and the consistency of the data was analyzed. Only two outliers had to be excluded. The database will be used for fitting the free parameters of the modality selection model derived in the next chapter.

Chapter 4
A Computational Model of Modality Selection

The aim of this chapter is the derivation of a computational model that enables the prediction of modality usage if more than one input modality is offered (Schaffer et al., 2015). Section 4.1 briefly summarizes the motivation for a utility-driven model. In Section 4.2, the model is created step by step. The predictive power of the model is analyzed in Section 4.3. A first application example is presented in Section 4.4. Section 4.5 concludes with a summary of this chapter.

4.1 Motivation

The empirical data presented in Chapter 3 implies that system users adapt modality usage to the estimated utility of modalities. Speech input is usually preferred, if it is more efficient in terms of interaction steps. In contrast speech usage decreases with increasing ASR error rate. The perceived utility, guiding users' modality choice, is affected by modality efficiency and input performance. The factors come along with a cost-benefit tradeoff: the expected utilities of modalities are offset against each other. In our model, efficiency is operationalized as the number of needed interaction steps to solve the task. If a task can be solved with fewer interaction steps in a specific modality, a shortcut exists, increasing the probability of modality usage. Further, input performance is operationalized by system errors like ASR errors or touchscreen malfunction. These factors are outlined in Table 4.1. The factor i is determined either by overt touch input or by utterances aimed at inputting information using speech input. An example of interaction steps considered by the model was depicted in Table 3.1. The steps partly correspond to the steps of a process model, as defined by KLM or GOMS (John and Kieras, 1996). A close affinity between the utility model and process models is worthwhile, as thereby the integration of the modality choice mechanism into AUE tools like CogTool (John et al., 2004) is supported. Process models mainly predict the time it takes to complete a task and are not directly able to predict modality selection. In a GOMS way of speaking, touch and speech input can rather be understood as different methods and the modality

prediction model could eventually be used as a selection rule. In the next section the modality selection model is created step by step.

Table 4.1 Factors relevant to the utility driven model.

Factor	Description	Domain
i	Interaction steps needed for solving a task or sub task using a specific modality	$i \in \mathbb{N}$
er	The average error rate of a modality	$er \in [0,1]$

4.2 Model Derivation

4.2.1 Expected Number of Interaction Steps

4.2.1.1 Probability Criterion for a one-step Task

As mentioned before, empirical evidence implies that system users weigh the expected number of needed interaction steps while concurrently considering alternative modalities. The user can easily anticipate the pure impact of interaction steps if the steps to solve the task are easily comprehensible. However, the user cannot be absolutely sure about forecasting the exact number of interaction steps, since more steps than expected may be necessary due to system errors. Therefore, the aim is to operationalize the expected number of interaction steps by means of both factors, specifically interaction steps and system errors. Thus, a criterion incorporating interaction steps and error rate is needed to capture the costs of using a certain modality.

At first, *one*-step tasks are considered, i.e., tasks solvable in only one interaction step. In the case of the RBA, selecting a specific list and browsing for an item within a list are defined as distinct sub tasks. *One*-step tasks are sub tasks where the required list item can be found on the first list screen. Thinking about possible system errors *er*, such a task can be conducted with 1 to n steps, as errors entail that system input has to be re-performed. The probability *p(n)* for solving the task after n steps is:

$$p(n) = (1 - er) \cdot er^{n-1} \tag{4.1}$$

If no errors occur ($n = 1$), *p(1)* equals the accuracy $1 - er$ of the input interface. If errors occur ($n > 1$), the error probability er has to be considered for each error-prone interaction. Thus, if n steps are needed, er takes effect $n - 1$ times until the goal is reached after the n^{th} step at which the accuracy $1 - er$ has to be attributed. Table 4.2 illustrates the formula. The first row indicates possible values of n. The second row illustrates how the calculation of probabilities changes with increasing

n. For every step to be added, er must be multiplied. The last two rows show two examples for possible values of er.

Table 4.2 The probability to solve a one-step task with n steps.

n	1	2	3	≥ 4
$p(n) =$	$(1-er)$	$\cdot er$	$\cdot er$	...
$er = 0.1$	0.9	0.09	0.009	...
$er = 0.3$	0.7	0.21	0.063	...

According to the probability axioms, the total probability of all possible events ($1 \leq n \leq \infty$) must equal 1. In this situation the limited number of screens in the RBA does not act as an upper bound of the sum. The upper bound only states the number of entries that can be misinterpreted by the system directly after another, which is by definition not limited.

$$\sum_{n=1}^{\infty} p(n) = 1 \tag{4.2}$$

Utilizing the convergence of the geometric series (Cgs.), one can prove that this is the case for the aforementioned formula of $p(n)$

$$\sum_{n=1}^{\infty} p(n) = \sum_{n=1}^{\infty} (1-er) \cdot er^{n-1} = (1-er) \cdot \sum_{n=1}^{\infty} er^{n-1} \overset{Cgs.}{=} (1-er) \cdot \frac{1}{1-er} = 1 \tag{4.3}$$

However, a limitation must be mentioned: in the case of occurring ASR errors, the system reaction can be diverse. ASR errors can lead to other system states from where it is not possible to proceed directly with the actual task. In these cases, more than one step may be necessary to start a new trial (depending on the system implementation and the error type). The proposed model in simplified terms assumes that a new trial is possible directly after each system error.

4.2.1.2 Expected Number of Interaction Steps for a one-step Task

For a task actually solvable with one step, a higher number of steps can be expected depending on the probability of system errors. If the number of actual interaction steps corresponds to a stochastic variable S_1, for a one-step task, the expected number of steps can be calculated by means of the expected value $E(S_1)$

$$E(S_1) = \sum_{n=1}^{\infty} n \cdot p(n) = \sum_{n=1}^{\infty} n \cdot (1-er) \cdot er^{n-1} \overset{Cgs.}{=} \frac{1}{1-er} \tag{4.4}$$

The last simplification of the above expression again results from the characteristics of the geometric series (Cgs.).

4.2.1.3 Expected Number of Interaction Steps for a i-step Task

In an i-steps task, the probability of system errors has to be considered in each step. Thus, the expected number of interaction steps S can easily be deduced from the one-step task equation. A task with i steps involves one-step i-times, resulting in the expected value

$$E(S) = E(S_1) \cdot i = \frac{i}{1-er} \tag{4.5}$$

4.2.2 Modality Utility

4.2.2.1 Objective Utility

If two input modalities are offered alternatively, the number of interaction steps to solve the task can differ from each other for each modality. In other words, counting in interaction steps, the utility of two modalities can differ. Thereby, the higher utility is attributed to the modality incorporating less interaction steps to solve the task.

An objective representation of a modality's utility U_o can be generated from the inverse of the expected value of interaction steps. U_o is then a function of interaction steps i and error rate er.

$$U_o(i,er) = \frac{1}{E(S)} = \frac{1-er}{i} \tag{4.6}$$

Assuming an ideal case with $er = 0$, the utility for a task with one interaction step equals 1

$$U_o(1,0) = \frac{1-0}{1} = 1 \tag{4.7}$$

The calculation example illustrates the maximum objective utility. With $i \in \mathbb{N}$ and $er \in [0,1]$, one can assume that $U_o \in [0,1]$. Utility decreases if either interaction steps i or the error rate er are increasing.

4.2.2.2 Modality Specific Weighting of Interaction Steps

The equation can further be extended by a weight that integrates modality-specific effects of interaction steps. The additional factor should thereby incorporate modal-

ity choice moderators like the average time per interaction step of a modality, cognitive load caused by using the modality or personal preferences.

$$U_t(i,er) = \frac{1-er}{i \cdot t} \tag{4.8}$$

As we assume that the weight correlates with the average task time, we choose the identifier t and determine its domain as $t \in \mathbb{R}^+$. The value of t will be derived by parameter fitting.

4.2.2.3 Modality Specific Weighting of System Errors

Errors caused by different input components of a system are differently perceived and rated by the users. Empiricism shows that touch screen errors have higher influence on modality choice than ASR errors (Schaffer and Minge, 2012). According to the integration of modality-specific effects of interaction steps, the effects of system errors can be considered by extending the objective utility equation with a weighting factor w (with $w \in \mathbb{R}^+$). The value of w will be derived by parameter fitting.

$$U_w(i,er) = \frac{1-er \cdot w}{i} \tag{4.9}$$

4.2.2.4 Perceived Utility

To obtain a utility function affected by modality specific effects regarding interaction steps and input performance, both factors have to be jointly considered. Based on probability theory, the utilities U_t and U_w can be seen as statistically independent (Chow and Teicher, 2003). Thus the joint utility equals the product of single utilities. Perceived utility is therefore calculated by multiplying the weighted utilities U_t and U_w. The resulting function for perceived Utility U is

$$U(i,er) = U_t \cdot U_w = \frac{1-er}{i \cdot t} \cdot \frac{1-er \cdot w}{i} = \frac{1-er-er \cdot w+er^2 \cdot w}{i^2 \cdot t} \tag{4.10}$$

We assume that free variables differ between systems and user groups. Thus, values have to be calculated for each system version and user group, to obtain valid predictions for modality choice.

4.2.3 Modality Usage Probability

To calculate the usage probability of a modality m_1, the utility U_{m1} is divided by the sum of utilities of the involved modalities. In this way, modality usage probability

P_{m1} for the modality $m1$ is constructed. Regarding two input modalities, P_{m1} integrates all considered variables and is thus a function of all error rates and interaction steps of all modalities.

$$P_{m1}(i_{m1}, er_{m1}, i_{m2}, er_{m2}) = \frac{Um1}{Um1 + Um2} \tag{4.11}$$

$$= \frac{i_{m2}^2 \cdot (1 - er_{m1} - er_{m1} \cdot w_{m1} + er_{m1}^2 \cdot w_{m1})}{i_{m2}^2 \cdot (1 - er_{m1} - er_{m1} \cdot w_{m1} + er_{m1}^2 \cdot w_{m1}) + c \cdot i_{m1}^2 \cdot (1 - er_{m2} - er_{m2} \cdot w_{m2} + er_{m2}^2 \cdot w_{m2})}$$

The free variable c integrates the factors t of both modalities ($c = t_{m1}/t_{m2}$). To enhance readability, we substitute arithmetic expressions in numerator and denominator and get our final form of the modality probability after rearranging as

$$P_{m1}(i_{m1}, er_{m1}, i_{m2}, er_{m2}) = \frac{1}{1 + c \cdot (b/a)} \tag{4.12}$$

with

$$a = i_{m2}^2 \cdot (1 - er_{m1} - er_{m1} \cdot w_{m1} + er_{m1}^2 \cdot w_{m1})$$

and

$$b = i_{m1}^2 \cdot (1 - er_{m2} - er_{m2} \cdot w_{m2} + er_{m2}^2 \cdot w_{m2})$$

4.2.4 Intermediate Summary

To summarize, a model to forecast the probability of modality usage was derived, based on the perceived utility of currently available input modalities. The perceived utility is determined by the expected number of interaction steps, which can be calculated as a function of input performance and modality efficiency. By incorporating performance- and efficiency-specific weights, the model is adaptable to new systems and different user groups. The model computes higher modality usage probability values if the usage of a modality leads to fewer interaction steps and maintains a lower error rate. In the next section the predictive power of the model is analyzed.

4.3 Analysis of Predictive Power

In this section the predictive power of the model described in Section 4.2 is evaluated. Implementation, model parameters, target values, and the evaluation procedure are described in Subsection 4.3.1. Subsection 4.3.2 depicts the results of the performance analysis of specialized models (built from ASR, touch, or mixed error conditions) as well as an integrative model (built from all conditions). The performance

of the integrative model on specialized data is further evaluated in Subsection 4.3.3. The results are discussed in Subsection 4.3.4.

4.3.1 Model Settings

4.3.1.1 Implementation and Parameter Fitting

The modality probability model is implemented in Python. The free parameters w_{m1}, w_{m2}, and c are fitted by means of the Sequential Least Squares Programming (SLSQP) solver of SciPy 0.11. In our case, SLSQP optimization minimizes the mean squared error between predicted values of speech usage and corresponding empirical target values.

4.3.1.2 Input Parameters and Model Data Sources

Table 4.3 outlines the data of the 6 different conditions gained from the three experiments described in Chapter 3. For model development the data from all three experiments was combined. The experiments were designed for minimizing inter-experimental differences. The database described in Section 3.5 was used for parameter fitting. Column n indicates the number of participants for each condition. According to the modality probability equation derived in Section 4.2, for each modality the model's input parameters are the simulated error rates (er_{speech} and er_{touch}) and the number of interaction steps (i_{speech} and i_{touch}) form the current state to the next (sub) goal. Due to speech shortcuts i_{speech} is constant, whereas i_{touch} varies from 1 to 6 to reach the (sub) goals. The parameter i_{touch} was varied in all experiments, and is available for each experimental condition. Therefore for each combination of i_{speech} and i_{touch} the probability of speech usage p_{speech} is also differentiated in six levels for each participant.

Table 4.3 Data used for the different conditions. The last four columns specify what data were used for each model.

Cond.	n	er_{speech}	er_{touch}	i_{speech}	i_{touch}	p_{speech}	SED	TED	MED	INT
T00, S00	1-16	0	0	1	1-6	$p_{n,1} - p_{n,6}$	X	X	X	X
T00, S10	1-16	0.1	0	1	1-6	$p_{n,1} - p_{n,6}$	X	-	-	X
T00, S20	1-12	0.2	0	1	1-6	$p_{n,1} - p_{n,6}$	X	-	-	X
T00, S30	1-13	0.3	0	1	1-6	$p_{n,1} - p_{n,6}$	X	-	-	X
T20, S00	1-11	0	0.2	1	1-6	$p_{n,1} - p_{n,6}$	-	X	-	X
T20, S20	1-12	0.2	0.2	1	1-6	$p_{n,1} - p_{n,6}$	-	-	X	X

Four different versions of the model were trained, each comprising different experimental conditions as data sources. The last four columns of Table 4.3 specify precisely what data were used for each model. The first three models are a specialized speech error-driven (SED) model, a touch error-driven (TED) model, and a mixed error-driven (MED) model. The fourth model integrates all conditions (INT model). The SED model is trained using data sets comprising speech recognition errors. The TED model as well as MED model training data comprises data sets with touch screen errors respectively mixed errors conditions.

4.3.1.3 Target Variables

The prediction models estimate the probability of users' modality choice. An individual modality usage profile is available for each individual test subject. However, the individual data may introduce a significant amount of noise in the prediction, as modality usage behavior strongly varies between participants. Furthermore, averaging over all participants of a condition may lead to a loss of information.

The experiments described in Section 2 were designed for collecting modality usage data for six different levels of benefit of speech (with B_{speech} amounting from 0 to 5) and six experimental conditions (differing in the error setting). We therefore decided to use the following two target variables:

- Individual subject predictions are obtained by averaging modality usage over the same levels of benefit of each participant, resulting in 80 individual modality usage profiles.
- Averaged predictions are obtained by averaging modality usage over the same levels of benefit and over all participants of a condition, resulting in averaged modality usage profiles for the 6 conditions.

4.3.1.4 Performance Evaluation

Performance has been analyzed for the distinguished target variables and two training cases:

- All cases (ALL): Within-data performance was analyzed using identical training and test sets (interpolation). The analysis was conducted for modality usage predictions of individual subjects as well as for averaged data. All cases are used as training data.
- Leave-one-out (L1O): Out-of-data performance was analyzed by performing leave-one-out cross-validation (extrapolation). For modality usage predictions of an individual subject, data from one user is omitted in the training. The respective data are taken for testing a model trained on the remaining $n-1$ users. For modality usage predictions of averaged data, data from one condition is omitted in the training, and the respective data is taken for testing a model trained on the remaining conditions.

The performance of the obtained models has been evaluated by means of the amount of covered variance R^2 and by the root mean squared error $RMSE$. Prediction power of the integrative model on specialized data (speech error, touch error or mixed error data) is evaluated by performance comparison with specialized models. The lower the degradation of the INT model performance for the specific data, the higher is the robustness that can be attributed to it.

4.3.2 Performance Analysis Results

The performance measures for all models are given in Table 4.4. For the within-data (ALL), R^2 amounts from 0.40 to 0.44 for individual subject target values and 0.90 to 0.95 for averaged target values. Best results are obtained for the MED model. According to our assumption, averaged data provides better performance than the prediction of individual subjects' modality usage. Individual modality-usage profiles can vary strongly, resulting in lower R^2. The $RMSE$ amounts from 0.21 to 0.24 for individual subject target data and 0.04 to 0.06 for averaged data. Higher variance of individual subject data causes higher error values. Lowest error values are obtained for the MED model.

Table 4.4 Performance on training (ALL) and independent test data (L1O) with the speech error driven (SED), touch error driven (TED), mixed error driven (MED) and integrative (INT) models. Values in bold indicate best performance.

Configuration		Training	Performance	
Model	Target		R^2	$RMSE$
SED	Individual subject	ALL	0.433	0.243
SED	Averaged	ALL	0.910	0.066
TED	Individual subject	ALL	0.400	0.220
TED	Averaged	ALL	0.936	0.044
MED	Individual subject	ALL	**0.435**	**0.211**
MED	Averaged	ALL	**0.948**	**0.042**
INT	Individual subject	ALL	0.407	0.232
INT	Averaged	ALL	0.899	0.063
SED	Individual subject	L1O	**0.414**	0.245
SED	Averaged	L1O	**0.883**	0.075
TED	Individual subject	L1O	0.349	0.229
TED	Averaged	L1O	0.054	0.167
MED	Individual subject	L1O	0.402	**0.218**
MED	Averaged	L1O	0.522	0.126
INT	Individual subject	L1O	0.388	0.235
INT	Averaged	L1O	0.865	**0.073**

When testing the models on unseen test data (L1O), the performance decreases in all cases. This result shows that the models are better in interpolating the training data than in extrapolating to unseen test data. Regarding individual subject target data, the lowest degradation is observable for the SED model, whereas for averaged data the INT model is on a par with the SED model. The considerable degradation of the TED model for averaged target values can be attributed to the small amount of available training data. For the cross-validation, averaging over two conditions results in only one test and training set. The performance decreases due to considerable differences between these two conditions. Although the lack of training data also has an effect on the MED model performance, the decrease in R^2 is considerably smaller. The lowest *RMSE* can be reported for the INT model for averaged target data and for the MED model for individual subject data.

4.3.3 Specialized Data Prediction Power of the Integrative Model

The integrative model is applicable to varying error conditions, as it is trained on all available data. However its capability in predicting special condition data may be limited. If only ASR and no touch screen errors arise, the predictive power of the integrative model can be reduced, as the touch error and mixed error conditions can have a contradictory impact on the fitting of free parameters used by the SED model. The performance of the specialized SED model may in this case be higher. However, with regard to conditions where more data is necessary for consistent model development, the inclusion of similar conditions might improve prediction performance. Therefore the predictive power of the integrative model on specialized error conditions is of interest. In Table 4.5 the performance values of the integrative and specialized models on specialized test data is compared.

Considering within data (ALL) comparisons, only marginal performance losses can be observed for the integrative model. The differences between averaged and individual subject target data are in line with the results from Section 4.3.2. The performance comparison on unseen data (L1O) reveals consistently increased predictive power of the integrative model. Significant gain can be observed for average MED and TED data. The inclusion of all available data here takes effect. If not enough specialized data is available, notable performance improvements are possible by employing the integrative model.

4.3.4 Discussion

Four models for predicting modality choice in differing error conditions were analyzed with respect to their performance in both describing known within-data (interpolation) and predicting unknown test data (extrapolation). For extrapolation leave-one-out cross-validation was performed. Further an integrative model was compared

Table 4.5 Performance comparison of the integrative and specialized models on training (ALL) and independent test data (L1O). Values in bold indicate best performance.

Configuration		Training	Specialized models		Integrative model	
Test data	Target		R^2	*RMSE*	R^2	*RMSE*
SED	Individual subject	ALL	0.433	0.243	0.432	0.241
SED	Averaged	ALL	0.910	0.066	0.909	0.066
TED	Individual subject	ALL	0.400	0.220	0.391	0.222
TED	Averaged	ALL	0.936	0.044	0.911	0.051
MED	Individual subject	ALL	**0.435**	**0.211**	0.426	0.213
MED	Averaged	ALL	**0.948**	**0.042**	0.929	0.049
SED	Individual subject	L1O	0.414	0.245	**0.416**	0.245
SED	Averaged	L1O	0.883	0.075	0.883	0.075
TED	Individual subject	L1O	0.349	0.229	0.366	0.226
TED	Averaged	L1O	0.054	0.167	0.841	0.069
MED	Individual subject	L1O	0.402	0.218	0.407	**0.217**
MED	Averaged	L1O	0.522	0.126	**0.850**	**0.071**

to three specialized models, regarding the predictive power on specialized error conditions.

Concerning the training on all available data (interpolation), all models show considerable fit on averaged target data. Significant performance degradation can be observed for all models when describing individual subject behavior. The effect can be attributed to differences in individual modality-choice profiles. Individual subject profiles sometimes show contradictory behavior compared to averaged data, resulting in decreased goodness of fit.

For extrapolation the SED model performs best with respect to both average and individual subject behavior. If only ASR errors are present in the data, average user behavior is generally predicted correctly. The extrapolation performance of the integrated model is similarly good. Considerable performance losses when compared to interpolation can be observed for the MED and the TED model. These losses can be attributed to the lack of available training conditions. Data for more differing touch error conditions as well as mixed error conditions are needed to build models that can enable improved predictive power for unseen data.

The examination of the predictive power of the integrative model on specialized error conditions revealed only marginal performance losses for within data. Testing the integrative model on unseen data revealed consistently increased performance for all conditions. For average MED and TED data an enormous increase of predictive power was observed, strengthening the reliability of the integrative model. All in all, the integrative model seems to be a useful means to predict modality choice for various error conditions. The remarkable extrapolation strength for unseen user behavior, in particular, constitutes a substantial value and supports the model's transferability capabilities and validity.

It can be expected that the integrative model would beneficially support simulation tools for multimodal HCI since interactive behavior should be reproduced more realistically than by randomly selecting modalities. Existing impairments in predictive power and prediction errors can be caused by factors influencing modality choice that are not covered by the model so far. In the field of AUE such factors can also bias the usability predictions generated from simulated interactions. It is possible that unconsidered factors have an effect on specific tasks, systems, or user groups. To gain knowledge about which usability evaluations should be treated with caution, it will be necessary to identify reliability gaps in the proposed model as well as the factors causing these gaps.

In the next section a first application of the model implemented in MATLAB gives an impression about an existing limitation of the model. An indication that the information that the users obtained from the task does not completely match their mental model of the system is detected. As a preparation of the later integration in real AUE tools, the example also demonstrates how the model can successfully be implemented within state-based simulations. In order to reveal significant effects of so far unconsidered factors of modality selection that might limit the applicability of the model detailed comparisons between the behavior of real users and simulated data have to be performed.

4.4 Application Example

Before the integration of the model into the real AUE tools MeMo and CogTool is described in Chapter 5, in this section the models' general applicability for state-based simulations is tested. The application example aims at demonstrating benefits and limitations of the developed modality choice mechanism before the more elaborate integration in AUE tools. System, user, and task models are implemented using MATLAB[1].

4.4.1 System Model

The MATLAB system model of the Restaurant Booking Application (RBA) introduced in Chapter 3 is implemented as a finite state machine, a concept often used as a computational representation of interface designs (Möller et al., 2006). Looking for a restaurant using the RBA, the main task is to select predefined list items. The list selections comprise the following sub tasks: (1) city, (2) cuisine, (3) time and (4) people. Using speech input all sub tasks are directly processable within the first list screen. As each list item is directly accessible by speech input, the interface of the system reveals speech shortcuts. In the GUI, all items were ordered alphanu-

[1] http://www.mathworks.com/products/matlab/

merically. In the numerical sub task (4) the systems' list for selecting the number of people runs from 1 to 24, and in sub task (3) the list to select the time runs from 12 (via 24 and 1) to 11. In the alphabetical sub tasks the number of items per letter was balanced (with two items per letter).

4.4.2 User and Task Model

The overall empirical data implied that the participants were able to easily derive information about the efficiency of touch screen input from the GUI. The number of steps to solve a sub task could be estimated from the anticipated sequence of system states required to pass. Therefore the MATLAB user model was provided with the following knowledge: the sequence of states to fulfill a sub task and, for each state, the number of interaction steps to finish the sub task by means of speech and touch screen input. For the simulation in MATLAB three tasks from the real experiment were randomly selected. Thereby, it was ensured that all experimentally varied levels of speech shortcuts (B_{speech}, compare Section 3.1.1.2) were covered. The simulated tasks and the respective sub goals can be seen in Figure 4.1. Referring to Section 4.2.3, the probability of speech input P_{speech} was predicted using $i_{m1} = i_{speech} = 1$ (constant number of speech interaction steps) and varying number of touch screen interaction steps $i_{m2} = i_{touch}$ (ranging from 1 to 6). The user model was equipped with the integrative modality choice mechanism trained with all available data (compare Section 4.3) and knowledge about the optimal path to solve a task[2]. The analysis of the empirical data revealed that the participants only switched modalities on the home screen and on the first list screen. Modality choice was therefore performed only at these screens during the simulation. In this simplified application example decreased performance of input modalities was not considered ($er_{speech} = er_{touch} = 0$).

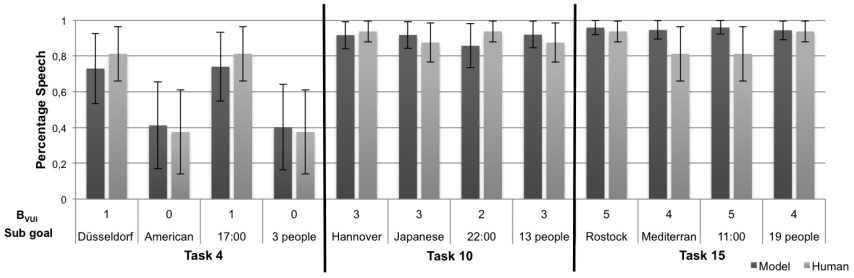

Fig. 4.1 The relation between B_{speech} and the percentage of speech usage. Each task contains four sub tasks. The tasks were randomly selected. However, it was ensured that each sub goal corresponds to a specific amount of B_{speech}.

[2] Due to the simplicity of the task almost no user errors occurred during the experiments.

4.4.3 Simulation Results and Discussion

Modality usage data gained from the MATLAB simulation was compared to human data gathered in experiment 1, which is based on the corresponding input parameters i_{speech}, i_{touch}, er_{speech} and er_{touch} (compare Section 3.2). Overall a considerable fit of the model to empirical data can be reported ($R^2 = 0.868$, $RMSE = 0.065$). For both human and model data, Figure 4.1 depicts the relation between the percentage of speech usage and B_{speech}, whereas the latter was varied between the sub goals[3]. The standard deviation of the model data shows the same relation as human data: the higher the percentage of speech usage, the narrower the standard deviation. The predicted percentage of speech usage is mostly in line with human data. Therefore, for most sub tasks, decisions made on the basis of the modality choice model are of use. In the application example, a system designer might draw the same conclusions from the model as from the human data, namely that the speech input is preferred by the user as soon as it becomes more efficient, and that the touch screen is selected more often if both modalities are equally efficient. This information could be used to change the interface design. The user could activate speech input if no speech shortcut exists. If speech shortcuts exist, however, speech input could be automatically activated and an appropriate notification could be integrated in the GUI.

However, human and model data are not totally in line. For the numerical sub tasks (”time” and ”people”) noticeable deviation can be observed for two cases: compared to model data, lower speech usage would be expected for human data in the sub goal ”22:00” of task 10 and higher speech usage in the sub goal ”11:00” of task 15. The model at this point strictly adjusts modality usage to B_{speech}. However the human decision seems to be influenced by additional information. Considering the information the user gains from the task, for both cases human speech usage data is associated with numerical values: for ”22:00” higher speech usage can be observed then for ”11:00”. Table 4.6 depicts the discrepancy between B_{speech} and numerical values within the list screens. It can be seen that the height of the numerical values on the list screens can not directly be associated with B_{speech}. This refers to an opposing effect of task and system design: in between tasks higher numerical values of sub goals are not necessarily associated with higher B_{speech}, as lists within the RBA start with different values (”12:00” for ”time” and ”1” for ”people”). For alphabetical sub tasks, a similar tendency could be observed. The observed speech usage points to the fact that the participants decision for a modality is affected by the alphanumerical information obtained from the set task. This provides an indication that the information that the users obtained from the task does not completely match their mental model of the system. However, it has to be noted that the described effects of task design were not systematically varied in the studies. Further research will be needed to investigate the significance of the observations.

The modality choice model is so far not able to cover the described task effects. As the model was build using overall data from tasks, including sub tasks with di-

[3] Note that the distribution of B_{speech} was well balanced over all of the 15 experimental tasks, although it is not absolutely balanced in task 10.

Table 4.6 Discrepancy between B_{speech} and numerical values within list screens.

B_{speech}	0	1	2	3	4	5
List screen	1	2	3	4	5	6
Numerical values	12	16	20	24	4	8
	13	17	21	1	5	9
	14	18	22	2	6	10
	15	19	23	3	7	11

verse combinations of B_{speech}, task effects should generally average out. An overall comparison between alphabetical and numerical tasks revealed no significant effects regarding modality usage. A system designer using the model to forecast modality choice behavior should confirm possible mismatches of task and system design and carefully select the tasks to simulate.

4.5 Chapter Summary

In this chapter a utility-driven model enabling the prediction of modality selection has been derived. Modalities are selected based on the perceived utility of currently available input modalities. The perceived utility is determined by the expected number of interaction steps, which can be calculated as a function of input performance and modality efficiency. The model incorporates performance- and efficiency-specific weights in the form of free parameters. These parameters were fitted to the empirical data gathered in Chapter 3. The analysis of predictive power reveals that a model integrating data from all experimental conditions may beneficially support AUE tools for multimodal HCI.

An application example demonstrates how information about simulated modality usage can be utilized to evaluate the design of a multimodal interface. Designers can use this information to adapt the interaction design of a system. Further an opposing effect of task and system design is uncovered: in between tasks higher numerical values of sub goals are not necessarily associated with higher B_{speech}. Instead, human speech usage data is associated with the numerical values, e.g. for selecting ”11:00” speech usage is relatively low although B_{speech} is at the maximum. This effect is not covered by the model. An application of the model for the AUE tools MeMo and CogTool is described in the next chapter.

Chapter 5
Automated Usability Evaluation of Multimodal Interaction

Several HCI studies revealed that multimodality affects the quality judgments of system users (Metze et al., 2009; Wechsung et al., 2009). The employment of specific modalities can lead to different experiences during the interaction. As a consequence, modalities should be selected as accurately as possible for automatic usability evaluation (AUE), estimating the quality of multimodal systems. This chapter documents the practical application of the created modality selection algorithm within AUE tools. Section 5.1 documents the work done in order to enable multimodal simulations with MeMo and exemplifies the creation of multimodal MeMo models and their application for the prediction of interaction steps. Accordingly Section 5.2 explains how ACT-R models exported from CogTool can be rendered multimodal and how the task execution time of skilled users with multimodal system designs can be predicted by means of these models. Finally a comparison between MeMo and CogTool is drawn in Section 5.3.

5.1 The MeMo User Simulation for Multimodal Interaction

In Section 2.3.6 it was concluded that multimodal systems can in principle be modeled with MeMo. However, human modality selection can not be simulated correctly as necessary mechanisms are missing. It can be assumed that the application of the algorithm derived in Chapter 4 represents more realistic predictions regarding modality selection. To integrate the algorithm, needed input parameters have to be available at the right time. Subsection 5.1.1 describes necessary extensions and new integrations enabling multimodal simulations. It was further argued that MeMo enables practitioners to create models of user interfaces and models of tasks to be solved by means of these interfaces. Subsection 5.1.2 illustrates how attributes of interfaces can be annotated, and how information exchange between user and system model as well as task knowledge can be modeled exemplified by the RBA. The RBA MeMo model will be the basis for the following sections dealing with the simulation. In Section 5.1.3 it is shown that the modality selection algorithm

is correctly implemented and that the necessary extensions do their work. Section 5.1.4 provides an application example for predicting the number of interaction steps to solve specific tasks. Section 5.1.5 provides an overall discussion of the MeMo modeling works.

5.1.1 Multimodal Extension

The module-based approach of the MeMo workbench and in particular the subdivision of the user model into several modules allow for the exchange and change of individual modules. Table 5.1 summarizes significant extensions and new integrations that have been made to the MeMo workbench. Within the components printed in bold most relevant changes were made. The following subsections focus on the description of these changes. The other changes are less complex. As they are part of the new simulation process their role and functionality are sketched together with the explanations of the most relevant changes.

Table 5.1 MeMo extension an new modules.

Change	Component	Description
Extension	Perception module	Implementations for the perception of multiple modalities
New	**Processing module**	In tegration of modality selection
New	**List tracking module**	A new module for the annotation of dialogs with interaction options that are part of a list
Extension	**Solution path calculator**	Implementations supporting the usage of multiple modalities
Extension	ASR error simulation	Reuse of the ASR error rate so that the error rate of the ASR is also taken for the modality selection
Extension	Logging	Expansions enabling the collection of modality choice data
New	Modality selection properties	File for the configuration of the modality selection parameters

5.1.1.1 List Tracking Module

The empirical results in Chapter 3 disclosed that it seems to be a prevalent strategy not to change the modality within a list. When a list screen of the RBA was entered, almost all users decided for a specific modality only at the first list screen and stayed in this modality if they browsed through the list. The list tracking module was developed to implement this strategy. It represents an extension of perception module. To make use of the list tracking the modeler can annotated dialogues as lists. During

the simulation a ”ListTracker” checks if a dialog is part of a list. While entering and leaving lists, it is examined if the user model should select a modality. According to the described user behavior the logic in Table 5.2 for performing this examination was created. The logic defines five cases in order to determine if the modality should be selected or not. In each case the possible changes between the current list and the previous list are taken into account. In the logic no/empty means that no list was defined or that a defined list is empty. In summary, the modalities are always selected, unless the user was in the previous state not in the same list as in the current state. The next subsection describing the integration of the modality selection processing module also refers to the list tracking and explains its application.

Table 5.2 Decision logic of the ListTracker: taking the status of the previous list and the current list into account, five cases depict how the ListTracker affects, if a modality selection is performed or not. In the first four cases where the previous list and the current list differ from each other a modality selection will be triggered. The fifth case describes that a modality selection will not be triggered if the previous list and the current list are identical.

Case	Previous List	Current list	Modality selection
1	no/empty	L1	yes
2	L1	no/empty	yes
3	no/empty	no/empty	yes
4	L1	L2	yes
5	L1	L1	no

5.1.1.2 Changes of the Solution Path Calculator

The following extensions were made:

Path calculation with multiple modalities: The path search has been extended by a modality filter for selecting the next steps. It is therefore e.g. possible that only ”speech steps” or GUI steps may be used for the calculation of solutions.

Search of an ”optimal” solution before simulation: The standard use of the path search is to search an ”optimal” solution before the simulation. For multimodal simulations in principle 3 optimal paths are calculated: a multimodal path, a path using the touch modality only, and a path using the speech modality only. For multimodal simulations the calculation of the optimal path is performed at each interaction step.

The determination of the steps to the goal: For multimodal simulations a determination of the steps to the goal has been added for individual modalities. The optimal solution is calculated starting from the current state.

Determination of partial solutions: A determination of partial solutions during the simulation has been integrated, to speed up the calculation of the optimal

solution. This was necessary because the exponential effort for the solution calculation increases considerably when using many sub tasks. In summary, only solutions for the next sub task are calculated by this extension.

5.1.1.3 Integration of the Modality Selection Processing Module

The modality selection processing module is derived from the default processing module. The single steps of the simulation of multimodal HCI are basically the same as described in Section 2.3. After the initialization the user model is equipped with task knowledge, and the system model is set in the start state. The existing perception module was changed in order to enable the perception of multiple modalities. All possible modalities are gathered now and then passed to the processing module.

Within the default processing module the use of multiple modalities was not supported. Once a possibility for speech input was found the speech path was selected, even if an interaction object for touch input was part of the actual state of the system model. The changes in the perception module now allow for performing the choice of a particular modality within the processing module. To overcome the limitations of the default processing module the following steps are realized in the modality selection processing module:

1. Check which modalities are available.

- if only one modality is available a selection is not necessary. The simulation continues with step 3.
- if two modalities are available, check if the user model actually navigates through a list or not.
  - if no, go to step 2.
  - if yes, go on with the last-used modality and continue the simulation (step 3).

2. Choose modality by decision algorithm.
3. Continue the simulation with the selected modality.

The steps described above illustrate how the modality selection processing module determines whether a modality selection should be performed. In the first step it is checked for which modalities solution paths exist. Solution paths can either be available for a single modality (GUI or speech input), or for both modalities. If a solution path is available only for one modality a selection is not necessary. The available modality is taken, step 2 is skipped and no modality selection is performed. The simulation continues with step 3. If solution paths exist for both touch screen and speech input, it has to be checked if the user model actually navigates through a list. Therefore the decision logic of the ListTracker in Table 5.2 is applied. If the user model navigates through a list in which it was previously, modality selection is not triggered (step 2 is skipped) and the simulation is continued with step 3. If the list was just entered or if the dialogue contains an empty list or no list, the modalities selection is triggered.

In step 2 the algorithm for modality selection derived in Chapter 4 is utilized. Performing the modality selection the input parameters, namely input performance and modality efficiency for all modalities currently having a solution, have to be made available. The input performance of speech input could easily be made available as it is already considered in MeMo for the simulation of ASR errors. However, for other modalities error rates are not part of the MeMo simulation. As argued in Section 2.5 the extension of the error simulation for other modalities is not being considered in this work.

Measured in interaction steps the efficiency of a modality to solve a specific sub task corresponds to the number of interaction steps of a partial solution of the task. For each modality, modality efficiency can therefore be calculated from the solution path to the task. The partial solution represents the steps from the current state to the goal state of the current sub task. As the current state and the goal state of the current sub task are known during the simulation, it is possible to calculate the partial solutions. Once the path of a partial solution is known, the number of involved interaction steps can be determined.

The modality selection is performed with the calculated modality efficiency for touch screen and speech input, and the specified input performance of speech input. For touch input perfect input performance is assumed. The algorithm computes the probability of speech input and decides for one of the modalities considering this probability. Accordingly, the return value of the modality selection is the result of the decision process, either to use touch screen or speech input.

Once the modality is fixed the simulation continues as described in Section 2.3. Interaction objects fitting well with the user knowledge are assigned with higher probabilities. As part of the processing, the rule engine can further influence the probabilities of the interaction possibilities of the selected modality. However, as depicted in the next section the rule mechanism is used for the simulations carried out in this work. In the last step of the processing module one interaction object is selected according to the probability distribution. The decision is passed to the execution module where the selected interaction is performed. Continuing the simulation, the system state is updated according to the user input and it is checked if a goal or a sub goal is reached. Accordingly, the interaction continues or the simulation terminates.

5.1.2 Modeling the Restaurant Booking Application with MeMo

5.1.2.1 System Model

The **dialogs** modeled for the RBA are listed in the dialog pool panel of the dialog designer shown in Figure 5.1. As an example the dialog ”City 1” is selected in which buttons to select a specific city and to browse through the list are modeled. In addition, the individual elements for speech interaction are listed. The graphical user interface of ”City 1” is shown to the right of the dialog pool panel in an editor

panel. The tab ”Voice” above the editor panel indicates that a speech interface is also created for this dialog. Under the ”Editor panel” is the ”Property Panel” in which various attributes of the dialog can be edited. Here also the new functionality for the perception of lists has been integrated. By utilizing the attribute *Containing List*, a dialog can be labeled as a list. All dialogs with the same label belong to the same list. For the RBA four lists have been created for the categories city, (culinary) category, time, and persons, each consisting of six dialogs (the list screens). Thereby the logic described in Table 5.2 is applied.

Together with the interaction objects for touch and speech input the **information** to be transferred to the system model is specified. For the RBA, this means that e.g. the variable *stadt* (engl.: city) is to be set to the value ”Berlin” for the Berlin button. For voice dialogs the information requested by the promt and the information that can be transferred to the system model via an attribute value pair has to be set. For all list screens this information transfer is straightforward. Figure 5.2 shows all variables that have been defined for the RBA. For the highlighted variable *stadt* the type string and a number of possible values are specified. The RBA information pool splits up into domain variables and system variables. All domain variables have type string and define either the possible values of the categories, or the category selected at the start screen. The system variables are used in the transitions. They have the type boolean and are used to define successful progress in solving the task or the end of sub tasks. The following description of states and transitions provides further insights into the use of information variables and the description of the RBA task models.

The RBA model consists of 26 **states** that are interconnected by speech and touch transitions. Since there are no interface elements that are to be re-used in multiple states, the modeled dialogs correspond to system states. In principle, all states follow the concept of touch screen and speech input presented here. Figure 5.3 shows a detail of the system model designer in which the state ”City 1” is is selected. In the graphical dialog on the left buttons are defined as interaction objects, whereas in the voice dialog on the right the interaction object for processing speech input is represented by the red arrow. The panel on the right of the dialogs contains all defined states and the panel at the bottom of the dialogs contains the transitions specified for the selected state. Views of the system graph can be looked up in Appendix B.1.

The graphical transitions are symbolized by ”OK” buttons, whereas the arrow symbolizes the voice transitions. If the domain variable *stadt* is set by touch or speech input of the user model, the system variable *isSetCity* is set to *true* through the consequence part of the transition. In Figure 5.4 left this is illustrated as part of a speech transition. In the condition part of the transition the parameters *! isNoMatch() and isValid()* are defined. *isNoMatch()* allows to specify the behavior of the system model in the case of ASR errors (deletions), whereas *isValid()* checks if the value of the requested variable is valid. The transition in Figure 5.4 right defines the system behavior if the parameter *isNoMatch()* is evaluated to *true*. As a consequence the system goes back to the start screen and no information is transferred to the system. As a consequence no information from the task knowledge is transferred to the

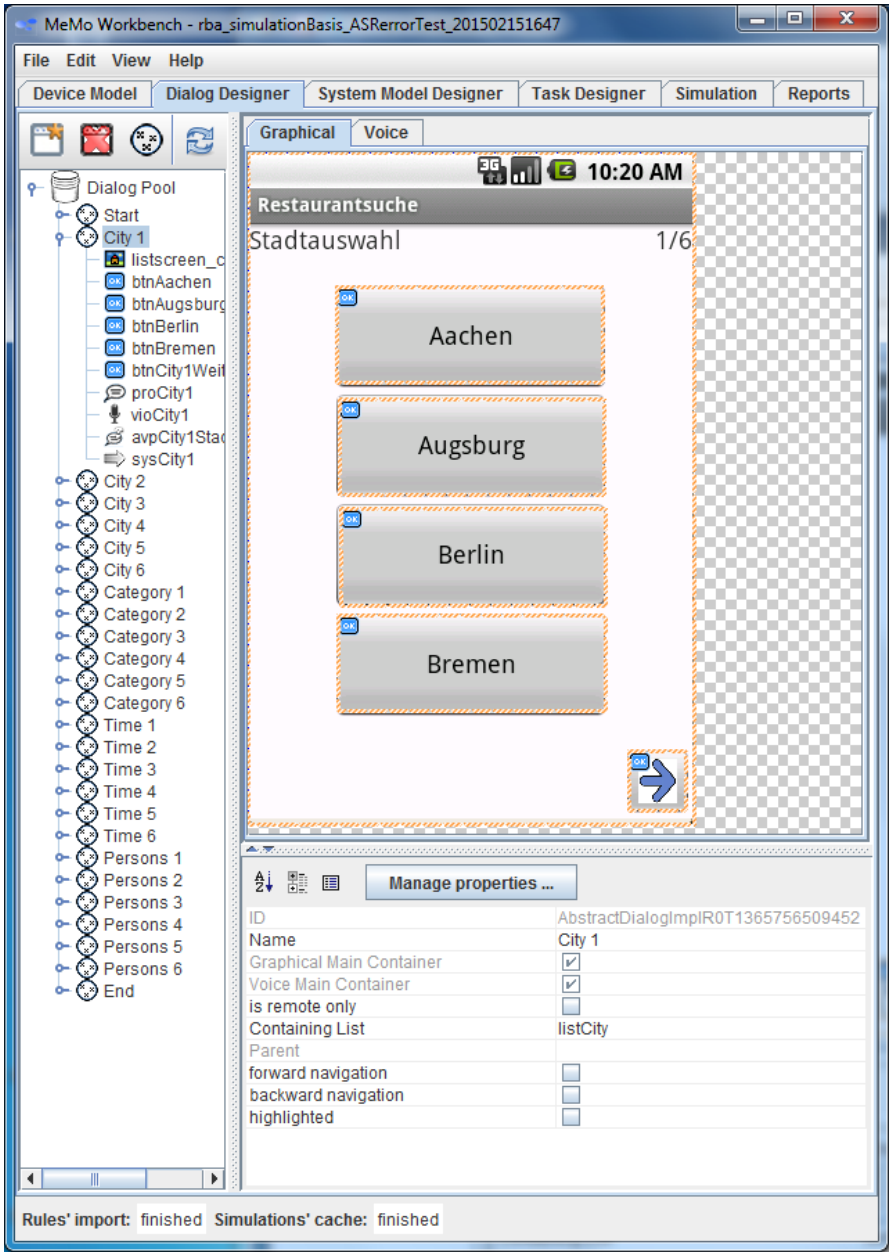

Fig. 5.1 Dialogs of the RBA.

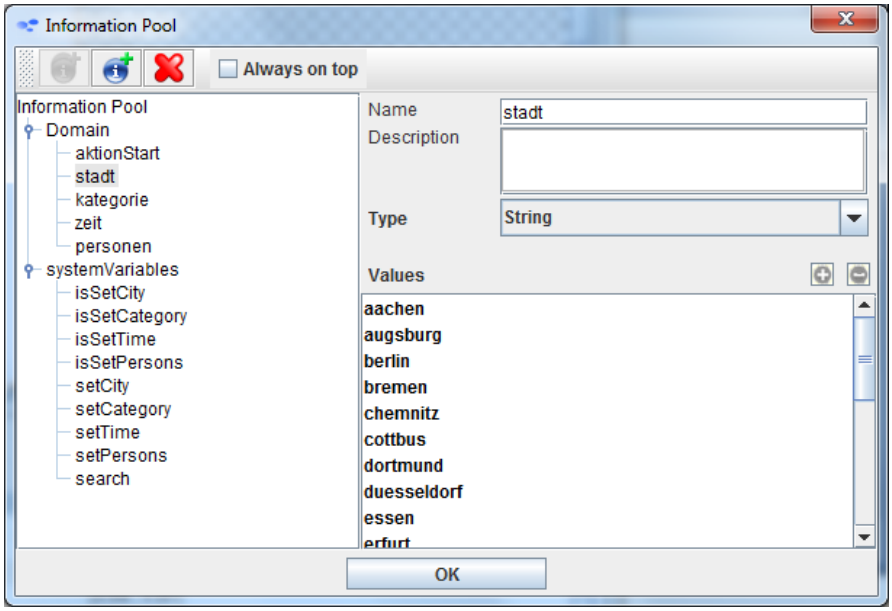

Fig. 5.2 Information pool of the RBA.

system model. Therefore the user model will interact with the system model again in order to meet respective goal conditions.

5.1.2.2 Task Models

As the entire restaurant selection task includes a relatively huge number of interaction steps and because the system model is compared to other MeMo models rather large, first simulation trials sometimes did not terminate. The available memory was overloaded. Therefore single tasks for each list depth (LD) were created. The performance of the entire tasks can then be composed of the results of the single tasks. The modeled tasks reflect the tasks of the empirical studies. For each of the investigated list depths exemplary task models are specified, meaning that for different tasks of the empirical studies with the same list depth only one representative task model (the LD task) exists. All of the six LD tasks are further split into two sub tasks. The first sub task is to select the required list at the start screen, and the second sub task is to pick the desired item from the list. The start screen is configured as start state.

Figure 5.5 shows the task designer. In the left panel the task named *LD5_SubtasksCityLeipzig* is selected. The middle panel shows the properties of the task. No specific initial system assignments (values for the available information variables) are initially set. The two sub tasks are listed in the lower part of the panel. The task properties for the selected sub task named ”SelectCity” are shown in the right panel.

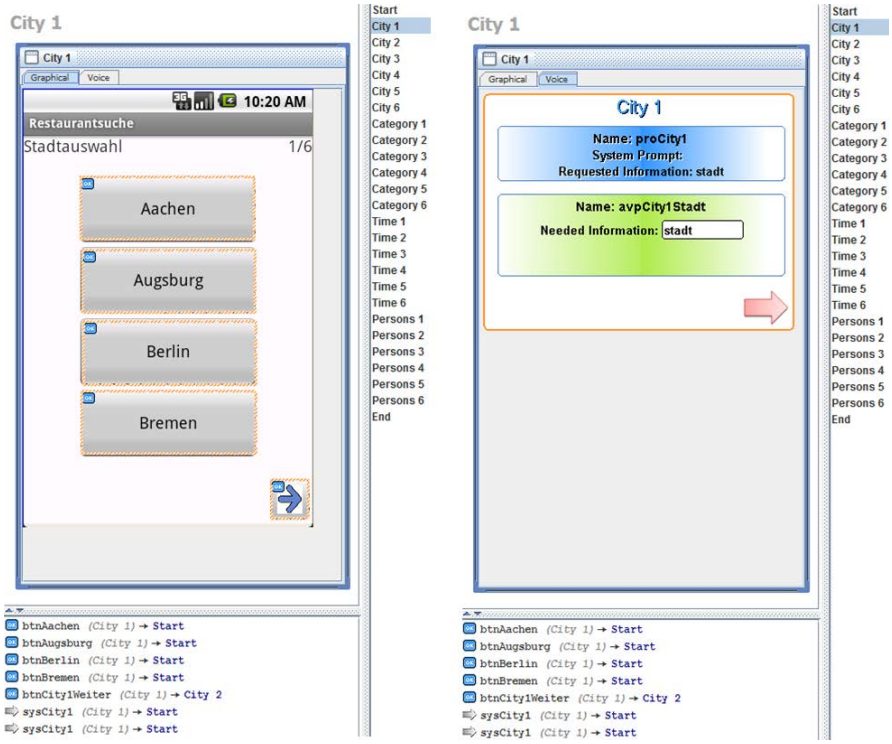

Fig. 5.3 The system model designer of the RBA.

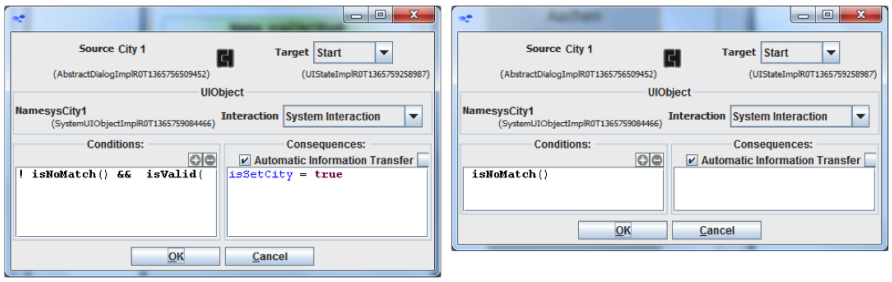

Fig. 5.4 Transition of the RBA.

In the user knowledge it is specified that the user model aims at selecting the city list as well as a specific city. Respectively the target values for the variable *actionStart* is set to ”stadt” and for the variable *stadt* to ”leipzig”. It has to be noted that the user knowledge of both sub tasks is the same. This ensures that the user model can continue with the execution of the task, if errors happened during the interaction. In the lower part of the panel the success conditions for the sub task are specified, including the condition *isSetCity == true* that ensures that the sub task has been conducted, and the condition *stadt == leipzig* that ensures that the right city has been chosen. All LD tasks are structured as described here. User knowledge and success conditions of all LD tasks can be looked up in Table 5.3.

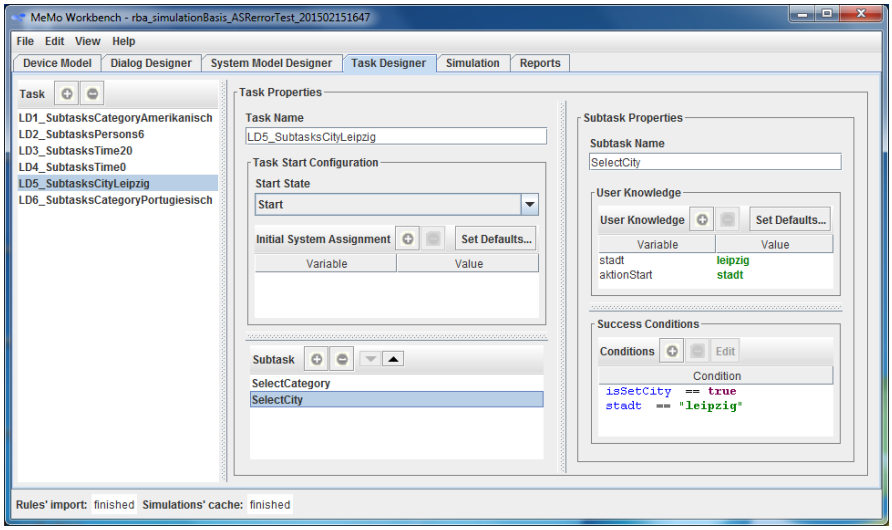

Fig. 5.5 The RBA task designer.

Table 5.3 User knowledge and success conditions of all LD tasks.

LD task	Knowledge - List (aktionStart)	Knowledge - List item
LD1	Kategorie	Amerikanisch
LD2	Personen	6
LD3	Zeit	20 Uhr
LD4	Zeit	0 Uhr
LD5	Stadt	Leipzig
LD6	Kategorie	Portugiesisch

5.1.2.3 User Model

Regarding the user model no specific attributes or limitations are required. Within the empirical studies the users were preselected in order to except users with specific limitations like physical impairments or language skills. The default user model of MeMo specifies standard values for all available user attributes. It can be looked up in Appendix B.3

Further no rules were used for all simulations, in order to eliminate any side-effects that could be caused by rules affecting the usage probability of single interaction objects. This can easily be done in MeMo by manually removing all the rules from the rules folder. The effect of this simple intervention is that the probabilities of the interaction objects are not affected by any rules, as no rules are existent.

5.1.3 Analysis of Simulated Modality Selection Behavior

The goodness of fit between human data and predictions of the modality selection algorithm was investigated in Section 4.3. In this section the integration of the modality selection model in MeMo is validated. The baseline for the analysis of the modality selection behavior of MeMo are therefore the predictions of the modality selection algorithm. These predictions serve as goal values for the modality selection model integrated in MeMo. MeMo decides for a modality according to the probability calculated by the modality selection algorithm. As a result of the MeMo simulation the simulated percentage of speech usage should be approximately the same as predicted by the algorithm. In order to validate the integration, in total 36 simulations were performed: 6 LD task simulations (differing in the level of modality efficiency) for each of the 6 experimental conditions of input performance. Each simulation contains 128 iterations. Consequently for each of the simulations log data for a specific LD task and a specific error condition is generated. Figure 5.6 illustrates the properties of the performed simulations. To adjust the error condition adaptions in the property files were made. The corresponding MeMo properties can be found in Appendix B.4.

In order to establish comparable data only simulated interactions performed at the first list screen of the RBA system model were considered. Therefore for this analysis the consequences of errors (referring to effects on state changes) do not affect the modality selection results, and the simulations can be performed for all conditions. In this context especially for touch screen errors it has to be mentioned the that modality selection behavior can be simulated, while a number of steps prediction can not be performed with the MeMo simulation so far as no mechanism for the consideration of the effects of touch screen errors is integrated.

The number of touch screen and speech inputs st the first list screen was counted and the percentage of speech usage was calculated. To illustrate the results of the simulations Figure 5.7 contains one diagram with the relevant LD results for each experimental condition. Data predicted by the modality selection algorithm (MSA)

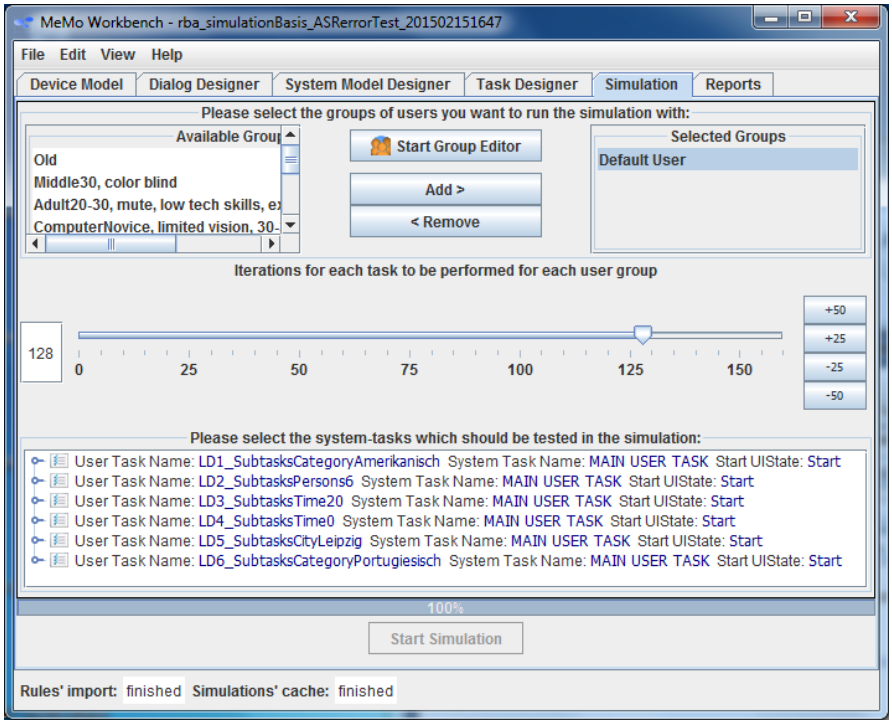

Fig. 5.6 Simulation view of MeMo.

is colored dark gray. All of the MSA curves have a smooth shape and the percentage of speech usage increases with an increasing benefit of speech. The effects of ASR errors can be best observed between the diagrams of the conditions [T00, S00] and [T00, S30]. If no errors occur and the benefit of speech is zero the probability to use speech already measures about 40 %, whereas the probability measures only about 20 % at the level of $B_{speech} = 0$ in condition [T00, S30]. With an increasing level of benefit this difference between the two error conditions becomes smaller and ends at a level of $B_{speech} = 5$ at about 5 percent. In the other conditions where only ASR errors occur, the two curves just described serve as lower and upper bounds. The percentage of speech usage always moves within these curves. If touch screen errors occur, a considerably higher speech usage can be reported. In the condition with touch screen errors only, speech usage measures already over 90 % if the benefit of speech is only one step. In the mixed error condition the increase of speech usage is smaller, but still considerably higher than in the other ASR error conditions. As in the human data reported in Section 3.4 touch screen errors are punished harder than ASR errors.

The data simulated by MeMo colored in light grey traces the curves of the modality selection algorithm quite well. Slight deviations arise as a result from the stochas-

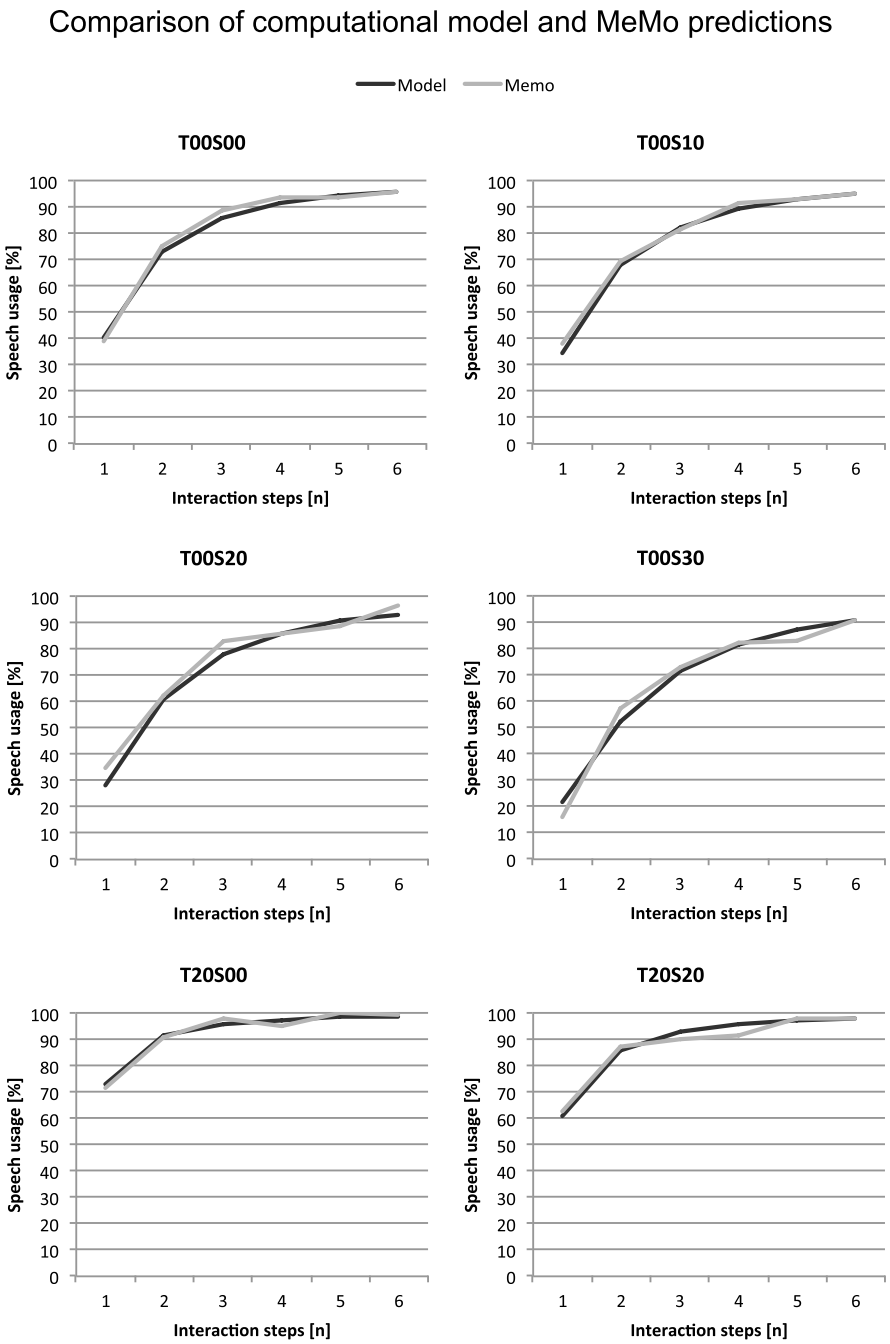

Fig. 5.7 Modality selection behavior of MeMo compared to predictions of the bare algorithm.

tic behavior of MeMo. The good fit is also reflected by a high overall R^2 of 0.98 and low *RMSE* of 0.03. The values for the single conditions are documented in Table 5.4.

Table 5.4 Performance measures of MeMo predictions for the validation of the integration of the modality selection algorithm.

Condition	R^2	*RMSE*
T00, S00	0.994	0.020
T00, S10	0.997	0.020
T00, S20	0.988	0.038
T00, S30	0.981	0.036
T20, S20	0.972	0.024
T20, S00	0.978	0.016

5.1.4 Application for the Prediction of Interaction Steps

In this section the simulation of tasks of the empirical studies with the RBA system model is depicted. The aim of the simulations is to predict the total number of interaction steps of these tasks.

5.1.4.1 Modelling specifics

Within MeMo it is not relevant if a sub task of the restaurant selection task is numerical (like "number of persons" and "time") or alphabetical (like "city" or "category"). As a consequence these kinds of tasks do not necessarily have to be distinguished in MeMo simulations. It must also be noted that with regard to modality selection no significant differences between alphabetic and numeric tasks performed by real humans could be found. For the prediction of the total number of steps to solve a task however, the list depth of the items searched is of considerable importance. As argued within this book users tend to adapt the input modality in order to save interaction steps, and thereby reduce the total number of interaction steps to solve a task. In order to test how the extended MeMo workbench performs the prediction three tasks were selected. The only limiting factor was that all levels of list depths should be included in the simulation. In order to produce comparable results it was decided to use the same tasks as in the application example depicted in Section 4.4, where the tasks 4, 10, and 15 were arbitrarily selected with the same requirements.

Regarding the different error conditions in human data, it has to be considered that it is so far not possible to simulate touch screen errors with MeMo. ASR errors

however, that can be simulated with MeMo, can affect the number of interaction steps when using speech input. A difference in the total number of interaction steps should be most evident, if the error conditions are most different. Therefore the conditions [T00, S00] and [T00, S30] were selected in order to verify the effect of errors. Human data from these conditions was used as a baseline for the simulation. The tasks 4, 10, and 15 were extracted from the log files and the number of interaction steps was counted for each subject and for each of the tasks. The arithmetic mean an standard deviation of interaction steps were calculated for all tasks in both conditions.

For the MeMo simulation the RBA system model with the the required error settings was used. Due to the memory issues already depicted in Section 5.1.2.2, LD tasks were simulated instead of the entire restaurant selection task. As a differentiation between alphabetical and numerical sub tasks is not necessary, the LD tasks depicted in Section 5.1.2.2 can be used as representatives of the actual sub tasks. Task 4 includes two sub tasks of LD 1 and two sub tasks of LD 2. Task 10 includes one sub tasks of LD 3 and three sub tasks of LD 4. Task 15 includes two sub tasks of LD 5 and two sub tasks of LD 6.

5.1.4.2 Simualtion and Prediction

The prediction of interaction steps comprises the following steps:

1. Simulate the four LD tasks suitable for the the actual task.
2. Calculate the average number of steps for each simulated LD task.
3. Calculate the predicted number of steps for the whole task.

Step one was performed with MeMo using 128 iterations per LD task. Before the simulation was started, the error properties were set, the default user group was chosen, and the respective LD tasks were selected. The settings can be viewed in Appendix B.4. The MeMo log file created from the simulation contains the single interactions for each LD task iteration. In step two for each iteration the number of interaction steps was extracted and the average as well as the standard deviation for the LD task was calculated. The predicted number of steps for the whole task was calculated by summing the partial results from the LD tasks. The standard deviation of the whole task was calculated by taking the square root of the sum of the squares of the standard deviations of the LD tasks.

5.1.4.3 Results

Figure 5.8 shows the results of the prediction of interaction steps, at the top for condition [T00S00], and below for condition [T00S30]. In general it can be seen that the mean values of human and simulated data correspond better if the standard deviations correspond well. Furthermore in both conditions a slight increase of interaction steps reveals in the human data between tasks. Here, it should be re-

membered again that the modality efficiency of speech increases between the tasks. Figure 4.1 summarizes the assignment of speech benefits to tasks. In the simulation data, the number of interaction steps increases from task 4 to task 10, but then decreases again in task 15. As this deviation of the simulation data is present in both conditions it can be assumed that the reason for the deviation is part of the model; respectively a factor that is so far not considered by the model shows effect.

In the condition [T00, S00] the number of interaction steps increases only slightly in the human data between tasks (with increasing efficiency modality of speech input). This can be attributable to the use of speech shortcut. Obviously the user tries to keep the number of interaction steps low. In the condition [T00, S30], an increased number of interaction steps can be observed in the human data of the tasks 10 and 15. This can be due to the fact that speech is used because it provides a shortcut, but at the same time ASR errors show their effect.

The large differences in the standard deviations of the model data occur if the user model selects the wrong list on the start screen. This can happen during the MeMo simulation with a low probability. Within the wrong list the user model does not find any information that could be provided to the system in accordance with its task knowledge. The forward and backward navigation is by default available in user knowledge. The interaction objects to move forward or backward therefore have a higher probability in the simulation than the other list elements of the wrong list for which no task knowledge exists. However, at some point a list item must be selected in order to return to the start screen. Thus, the user model browses through the list until a list item is selected. Due to the high probability of the forward and backward buttons, the probability that the user model stays within a wrong list for a certain time is relatively high[1], whereby the number of interaction steps can get large. Once the user model enters the start screen again by selecting one of the list items of the wrong list, the execution of the sub task can be continued. If large differences or in general high standard deviations appeared, the corresponding log files generated during the experiments with real humans or by the simulations were examined for outliers. The outliers occur either by remaining in a forward-backward loop in the simulation or by differing interaction strategies of real users.

The concrete values are reported in Table 5.5. If outliers occur, also corrected values are shown. For the simulated data corrections are done by removing the values of the respective iterations. In task 4 of condition [T00, S00] the predicted mean (M) and standard deviation (SD) of interaction steps match the human data quite well. The slightly higher standard deviation of the model data is caused by two rather small forward-backward loops each with 15 steps. The corrected values still fit fairly good. The large deviation of SD in task 10 is caused by a long forward-backward loop of 125 steps. After the correction both M and SD of the model data match the human data quite good. In task 15 high SD can be observed for human data. It was mainly caused by one user who made use of a interaction strategy that is not covered by the modality selection model, namely making use of touch screen only. Data by this user have been omitted for the corrected results. Two other users

[1] The author calls this a ”forward-backward loop”

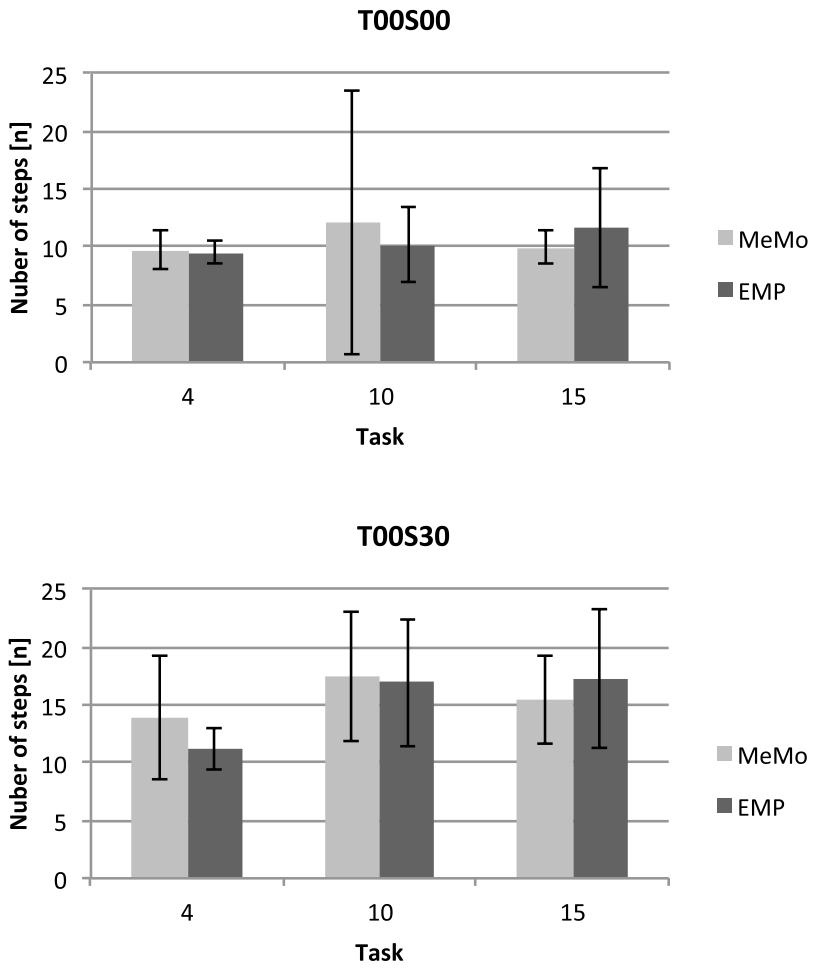

Fig. 5.8 MeMo prediction of interaction steps: comparison of empirical data (EMP) and simulated data (MeMo) in steps for tasks 4, 10, and 15. Error bars show standard deviation.

used speech quite rare and not as expected. However, as a clear strategy could not be identified for these users their data was not removed. Overall the corrected data therefore still shows some deviation, but the previous difference could be eliminated to a large extent.

In task 4 of condition [T00, S30] higher M and SD of the model data can be observed. The reason can be attributed to six forward-backward loops (with 11-47 interaction steps). After the correction the model data matches human data much better. In task 10 similar M but also high SD can be observed for human and model

Table 5.5 MeMo vs Human data.

Condition	Data	Task	M	SD	M (corr)	SD (corr)
T00, S00	Human	4	9.500	0.935		
	MeMo	4	9.677	1.711	9.283	0.619
	Human	10	10.125	3.238		
	MeMo	10	12.109	11.408	10.191	3.655
	Human	15	11.563	5.160	10.400	2.603
	MeMo	15	9.953	1.398		
T00, S30	Human	4	11.154	1.791		
	MeMo	4	13.9375	5.363	11.979	1.612
	Human	10	16.923	5.498	15.364	4.457
	MeMo	10	17.423	5.590	15.906	3.450
	Human	15	17.308	6.018	15.000	4.837
	MeMo	15	15.391	3.784	14.703	2.981

data. The high SD in human data occurred as a few users utilized nearly only touch screen, resulting in a higher number of interaction steps. Data of two users (with 25 and 26 interaction steps) have been omitted for the corrected results. The MeMo simulation came in some iterations into a forward-backward loop. After removing 4 iterations (with 18-43 interaction steps) the corrected values for M and SD still show a fairly good match. In task 15 the SD is relatively high within human and model data, whereas the M value for human data is considerably higher. Again one of the real users used touch screen only; further two users experienced an above average number of ASR errors, resulting all in all in a high number of interaction steps. These three users have been omitted for the corrected results. With regard to the simulation data 3 forward-backward loops with 14-25 interaction steps occurred. After the correction the model data match the human data better.

5.1.4.4 MeMo Reports

Additionally to the result for the simulated number of interaction steps for each performed iteration, MeMo delivers a report (see also Section 2.3.5). The aim of the report functionality is to provide insights into the usability of the tested system model. Regarding the multimodal simulation the possibility of displaying and browsing through the paths of the single iterations provides details about the involved system states and transitions. For each transition details like probability, conditions and consequences and alternative transitions can be viewed.

Figure 5.9 shows an example of the graphs of 4 iterations. All graphs have been generated by the MeMo simulation. The task with list depth 5 was to select the city "Leipzig". The labeled boxes symbolize system states and the green or black arrows with labels symbolize transitions. The transitions are green as long as the user model is on the optimal solution path. If this path is left, the transitions are black. The labels represent the name of the transitions, wheres the names have meaningful prefixes.

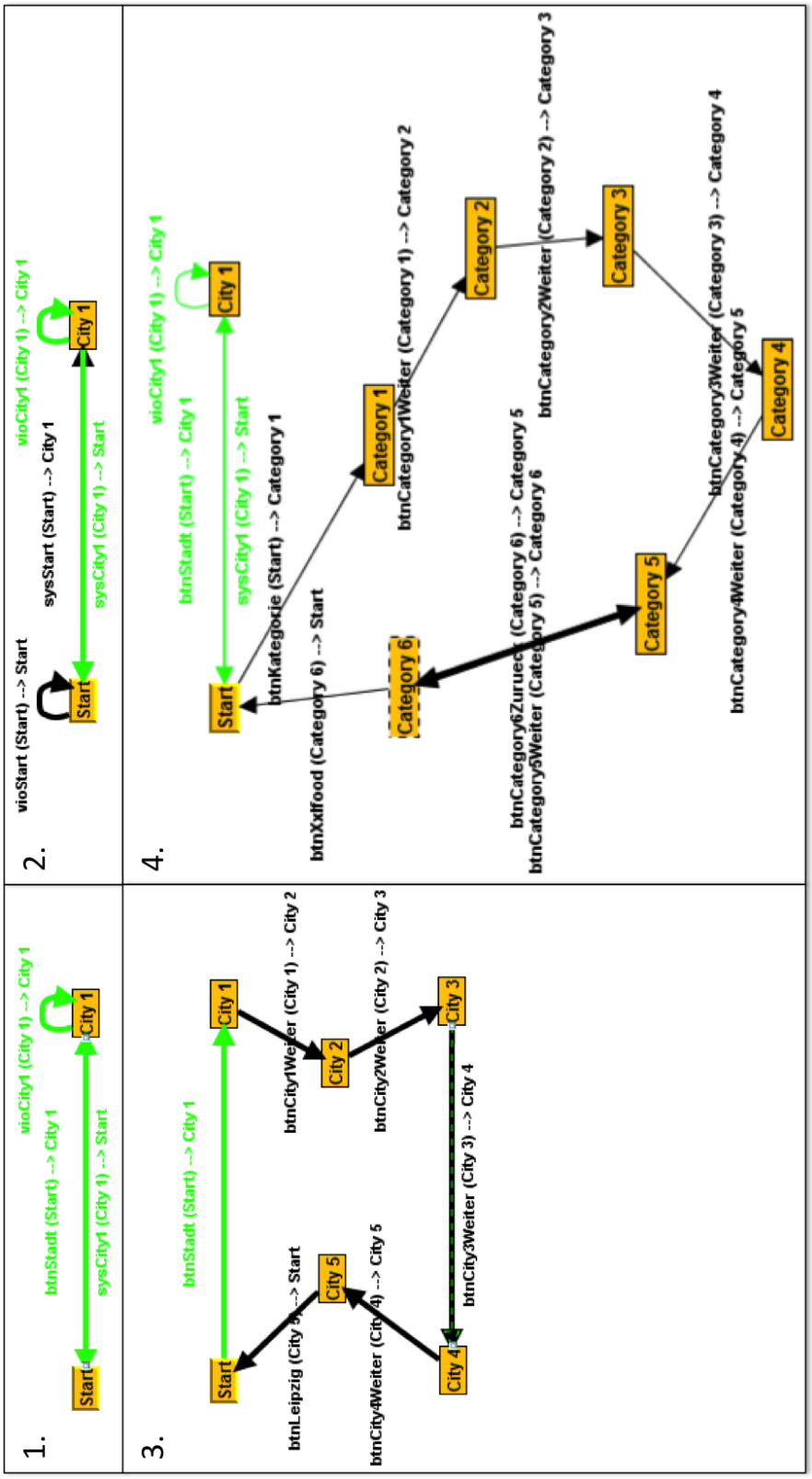

Fig. 5.9 MeMo simulation graphs.

The prefix ”btn” means that the corresponding interaction object is a button. The prefixes ”vio” and ”sys” represent the speech interaction objects, whereas a specific ”vio” transition always belongs to a specific ”sys” transition. The ”vio” transitions are easily distinguishable as they are always self-transitions of a system state. They act, so to speak, as the speech processing unit of the system. Information that is transferred from the user model to the system model is at first processed by the ”vio” transition and then the appropriate ”sys” transition changes the system state. In this way, also ASR errors can be generated[2].

The graphs 1,2, and 3 illustrate possible combinations regarding modality usage. Graph 4 is an example of an interaction error including a froward-backward loop. In graph 1 mixed modality usage is illustrated. In the state ”Start” at first the button *btnStadt* to select the city list is pressed. In the state ”City 1” speech input is used (with *vioCity1*and *sysCity1*) to select a city. All transitions are green because this mixed modality path also represents the shortest solution path for this task. Graph 2 looks similar with the difference that only speech input is utilized by the user model. Graph 3 illustrates the path if only touch screen input is used. The user model browses to the state ”City 5” where it finds *btnLeipzig*, the button to select the city ”Leipzig”. In graph 4 an interaction error takes place at the first interaction in the state ”Start”. Instead of pressing the button for the city list the user model presses the button for the list of the (culinary) category (btnKategorie). The MeMo report discloses that the probability for the right button would have been over 95%. However a low probability of choosing a wrong button that is always present in MeMo takes effect in this case (see. Appendix B.8 for concrete probability values). Within the category list the user model can not find an interaction object to transfer the right information to the system model. By means of the integrated search strategy it can browse to the end of the list. In state ”Category 6” the list ends and the backward button gets the highest probability. Back at state ”Category 5” an intrinsic preference of browsing forward lets the user model press the forward button again. The report revels that this loop has been carried out 55 times. Within the graph the thickness of the involved transitions also visualizes their high usage frequency.

As the optimal path is highlighted in green it is easy to notice when the user model leaves the optimal path. The report further shows that the optimal path is dependent on a particular combination of modalities. Certain users strategies, like using touch screen only may significantly increase the number of interaction steps. Simulating a sufficient number of iterations, one can assume that sooner or later almost every possible interaction path, including the various options of modalitiy combinations are found. Solutions that may involve usability problems are usually easy to recognize, as they considerably differ from the optimal path.

[2] It has to be noted, that the ”vio” transitions are not relevant for the calculation of modality efficiency for the modality selection algorithm. However they are considered separately within the report when the shortest solution path is calculated.

5.1.5 Discussion

The extensions made to enable the simulation of multimodal interaction between the user model and the system model work quite well. The analysis of the simulated modality selection behavior of MeMo revealed that the modality selection algorithm is correctly implemented into the workbench.

Though most other existing MeMo models include less states, compared to real systems the RBA model with its 26 states is not really big. However, the simulation of the whole restaurant selection task was not possible due to memory issues. By splitting the whole task into smaller parts this problem could be solved. To permit not only the analysis of smaller usability questions, but also to enable the modeling of large systems and complex tasks, it is desirable that these problems are solved MeMo internally.

The corrected predictions of the total number of interaction steps of three tasks with different modality efficiency provide useful results for two different error conditions. In combination with the MeMo reports the prediction results provide valuable insights into the usability of multimodal interaction. The reports reveal realistic modality usage as well as different possible interaction strategies. If interaction errors occur the user model can get into system states that are not on the optimal path. In real systems such errors can for example occur due to typing errors. In the subsequent states, the user model may have no information to transfer to the system. In MeMo thereby forward-backward loops can occur, causing and increased average number of interaction steps along with increased standard deviation. Through the distorted behavior of the user model an interaction problem can be detected. In this case, a back button could be helpful allowing to leave the list without selecting a list element. MeMo includes rules supporting such an error recovery.

After the corrections in both error conditions the human data interestingly shows a lower amount of interaction steps in task 15 then in task 10. As depicted above the same trend was observed within the model data. One can therefore assume that the differences in the uncorrected data arise from the interaction errors of the model on the one side, and from different interaction strategies of the users on the other side. Certain user strategies, like using touch screen only can significantly increase the number of interaction steps. After the corrections the model data reflects the data of human users quite good. This indicates that in addition to the modality selection strategy, other strategies such as monomodal system usage should be considered in MeMo simulations. Speech objectors could be modeled relatively easy by ignoring the speech input possibilities. The number of objectors could be derived from the data. For this work the consideration of the corrected data is acceptable as the investigation of different user strategies is not part of the studies conducted. The next section explains how ACT-R models exported from CogTool can be rendered multimodal and how the task execution time with multimodal system designs can be predicted by means of these models.

5.2 Adapting CogTool Simulations for Multimodal Interaction

As depicted in Section 2.4.6 multimodal designs and task demonstrations are possible with CogTool. However, if differences between tasks including divers sequential combinations of input modalities are to be explored, several alternative task demonstrations have to be recorded for each available modality combination of interest. The modality selection algorithm can be used as a mechanism automatically generating these combinations. The aim of this section is to test the feasibility and validity of a multimodal simulation that is based on ACT-R models generated by CogTool. Subsection 5.2.1 describes the procedure developed in order to enable multimodal simulations with ACT-R. Subsection 5.2.2 illustrates how the CogTool designs and task demonstration of the RBA were constructed. The RBA CogTool model will be the basis for the following sections dealing with the simulation. In Section 5.2.3 it is shown that the modality selection algorithm is correctly implemented and that the multimodal procedure in principle works. Section 5.2.4 provides an application example for the prediction of the total task execution time of specific tasks. Baseline predictions are generated with CogTool and ACT-R predictions are generated according to the multimodal procedure. Section 5.2.5 provides an overall discussion of the CogTool modeling works.

5.2.1 Multimodal Procedure

The aim of the multimodal procedure is to fuse a touch screen ACT-R model and a speech ACT-R model generated by CogTool into one multimodal model. The basic principle includes 2 steps:

1. **Model creation:** the CogTool part of the work, namely building a design of the system, demonstrating tasks, and exporting unimodal ACT-R models for each modality.
2. **Model modification:** building the multimodal ACT-R model.

An abstracted excerpt of the ACT-R model is illustrated in Figure 5.10. Regarding the RBA this example could represent the selection of a list at the start screen and an interaction on a list screen. Ovals symbolize concrete productions, while boxes include several productions. Both the touch screen solution at the left and the speech productions in the middle are generated from the CogTool script. The ”think” modality selection productions are manually integrated during the task demonstration. The changes on the arrows between the ovals and boxes on the right are realized by changes to the values of the goal state slots of the productions. Details of this procedure are depicted in the following two subsections.

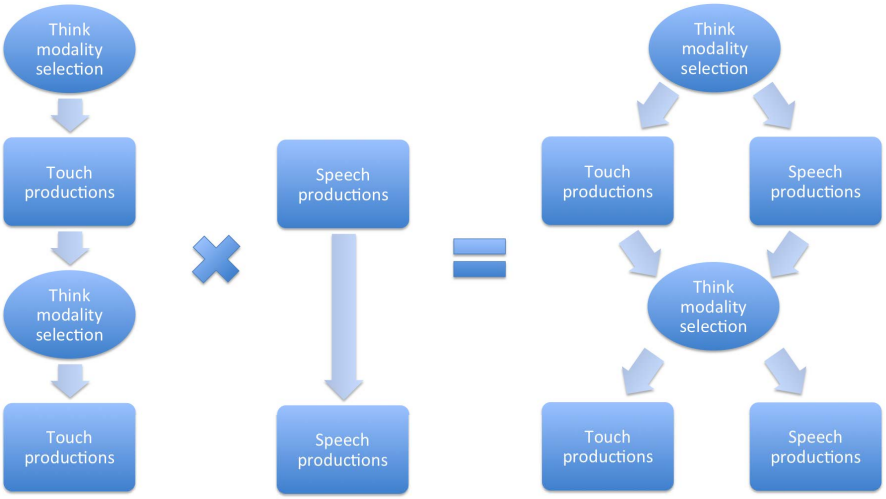

Fig. 5.10 Fusion of two unimodal touch and speech ACT-R models into one multimodal ACT-R model.

5.2.1.1 CogTool Model Creation

A design of the system has to be build with CogTool as depicted in Section 2.4.1. The design must include the graphical transitions and the speech transitions to solve the tasks of interest. Next, unimodal task solutions have to be demonstrated. CogTool supports to insert additional ”think” steps. As the touch solution will serve as the basis to build the multimodal ACT-R model additional ”think” steps are added at each CogTool frame of the touch solution where later the input modality should be selected. Together with the whole CogTool script these additional ”think” steps are translated into ACT-R code. An example of a ”think” step can be viewed in Listing 5.1. The duration of the additional ”think” steps is set to 0.1 seconds. Ideally, the duration should be 0 seconds because it is used only as a switch for the modality selection. However, in CogTool a ”think” step must have a duration greater than zero. Other ”think” steps are automatically inserted by CogTool before each speech or touch screen input. Examples can be viewed in the Listings 5.2 and 5.3. CogTool sets the duration of these modality dependent ”think” steps to 1.2 seconds. As it was found that these default values did not provide good predictions for the RBA system, modality specific values were calculated from the empirical data of the RBA experiments. The adaptions of the durations are depicted in the next section. After the tasks have been demonstrated for touch and speech input the ACT-R models can be exported. For each task the result are two files in the Lisp programming language, one for touch screen input and the other for speech input. These Lisp files can be run independently from CogTool within the ACT-R cognitive architecture.

5.2.1.2 ACT-R Model Modifications

The steps in which the modality selection should be executed have to be modified. As the duration of the "think" modality selection steps that were integrated during the task demonstration was set to 0.1 second, they can easily be identified. Within CogTool and within the generated ACT-R code these productions are the only ones with the action time of 0.1 seconds. Listing 5.1 shows an example for an "think" modality selection production.

Listing 5.1 "Think" modality selection ACT-R production generated from the CogTool script

```
1  ;; Script step "think" modality selection
2
3  (p Think-73
4    =goal>
5      isa klm
6      state 54
7    ?manual>
8      state free
9  ==>
10   !bind! =mod-selected (select-mod 1 2 0 0)
11   !bind! =state-mod (concatenate 'string "modspec8-1-" =
         mod-selected)
12
13   !eval! (set-cogtool-plist :duration 0.1 :current-frame "23
         _listscreen_persons_level1")
14   =goal>
15     state =state-mod)
16
17 (spp Think-73 :at 0.1)
```

The lines 10 and 11 are added to the right hand side of the production. *select − mod* is the function call for the modality selection algorithm. The Lisp implementation of the function can be looked up in Appendix C.1. The input parameters of the function are from left to right:

i_{speech} the number of interaction steps needed via speech input. In this work the number of speech interaction steps is always 1, as only speech shortcuts are considered

i_{touch} the number of interaction steps needed via touch screen input. In this example the number of touch screen interaction steps is 2.

er_{speech} the speech error rate. Speech errors are only considered for the analysis of simulated modality selection behavior depicted in Section 5.2.3. In this example the speech error rate is 0.

er_{touch} the touch screen error rate. Speech errors are only considered for the analysis of simulated modality selection behavior depicted in Section 5.2.3. In this example the touch screen error rate is 0.

In the CogTool/ACT-R examples the factors er_{speech} and er_{touch} are only considered for the analysis of simulated modality selection behavior depicted in Section 5.2.3. As argued in Section 2.5 extensions with regard to error simulations are not

being considered in this work. For the prediction of total task duration depicted in Section 5.2.4 conditions with speech and touch screen errors are not considered.

The factor i_{touch} is varied in the CogTool/ACT-R examples for the prediction of total task duration. The return value of the function is stored in the variable $=mod-selected$. It can be "speech" or "touch". The variable $=state-mod$ binds the value of the state slot of the next production. For the multimodal procedure the values of the state slots of the goal buffer need to be changed as now two modality specific solution paths must be considered. For this feasibility example "$modspec8-1-$" as the front part of the string is manually adapted for each modality selection production and also for the subsequent modality specific productions. After concatenating the front part of the string with the return value of the modality selection, the value of $=state-mod$ can in this example be either be "$modspec8-1-speech$" or "$modspec8-1-touch$". The numbering is so because in each task occur nine modality selections (here 8 was picked as a random example), each having subsequent modality specific productions (therefore, the subsequent numbering). The first modality specific productions for speech and touch are shown in Listings 5.2 and 5.3. In addition to the modified state slots, an !*eval*! statement can be seen on the right hand side of the productions. It is used by CogTool to detect the durations of productions and to characterize the current system state.

Listing 5.2 "Think" speech execution ACT-R production generated from the CogTool script

```
1  ;; Script step "think" speech execution
2
3  (p Think-32-speech
4    =goal>
5      isa klm
6      state "modspec8-1-voice"
7    ?manual>
8      state free
9  ==>
10   !eval! (set-cogtool-plist :duration 1.9 :current-frame "23
         _listscreen_persons_level1")
11   =goal>
12     state "modspec8-2-voice")
13
14 (spp Think-32-speech :at 1.9)
15 .
16 .
17 .
```

From the exported speech ACT-R model speech productions are copied and pasted into the former touch screen only ACT-R model. The durations in the modality specific "think" productions have to be adapted. In order to utilize realistic durations for the preparation of speech input and touch screen input the durations of real users at the first list screen were analyzed. For both modalities the average time on the first list screen was extracted from the log files of the experiments. Using touch screen humans stayed an average time of 1.3 seconds ($SD = 0.5$) on the first list screen; using speech input humans stayed an average time of 2.7 seconds ($SD = 0.7$) on the first list screen. It has to be mentioned that these times include

both a reasoning process and the execution of speech or touch input. The execution of speech and touch input is covered by making use of the ACT-R speech and motor modules. This execution always takes some time dependent on the length of the utterance (when using speech) or the actual position of the hand and the characteristics of graphical UI elements (according to Fitts' law). As a consequence the time for the reasoning process should be shorter then the average time at the first list screen. The best values for optimizing the prediction for the RBA design could be calculated by parameter fitting. However, this would only lead to an adjustment to the RBA, and can not be generalized for arbitrary systems. A first estimation for the time for the modality specific reasoning processes is therefore made by subtracting the standard deviations from the means. Considering that the "think" modality selection production involves a duration of 0.1 seconds, a duration of 0.7 seconds is obtained for the "think" touch production and a duration of 1.9 seconds for the "think" speech production. The productions in the Listings 5.2 and 5.3 include these times.

Listing 5.3 "Think" touch execution ACT-R production generated from the CogTool script

```
1  ;; Script step "think" touch execution
2
3  (p Think-74-touch
4    =goal>
5      isa klm
6      state "modspec8-1-touch"
7    ?manual>
8      state free
9  ==>
10   !eval! (set-cogtool-plist :duration 0.7 :current-frame "23
         _listscreen_persons_level1")
11   =goal>
12     state "modspec8-2-touch")
13
14 (spp Think-74-touch :at 0.7)
15 .
16 .
17 .
```

After the depicted adaptions the multimodal ACT-R model can be run using the ACT-R cognitive architecture. In these steps where the modality selection process has been integrated modalities are selected according to the specified input parameters. In the other steps originating from the CogTool script the predictions are performed as before. By running the multimodal model several times task solutions differing in modality selection can be produced. For all these solutions an average task duration with a standard deviation can be calculated.

5.2.2 Modeling the Restaurant Booking Application with CogTool

5.2.2.1 System Design

An overview of all frames modeled for the RBA can be looked up in the CogTool design window in Appendix C.2. Figure 5.11 shows details of the frame ”City 1”. Two buttons are modeled, one colored orange for browsing through the list of cities, the other one colored blue (highlighted) for selecting the city ”Aachen”. As depicted in Section 2.4.1 it is not necessary to integrate input options for interactive elements that are not used in the task demonstration. CogTool makes predictions of task execution time for skilled users, therefore only these elements have to be modeled that are definitely used. All other graphical system states (frames) are modeled similarly as in this example.

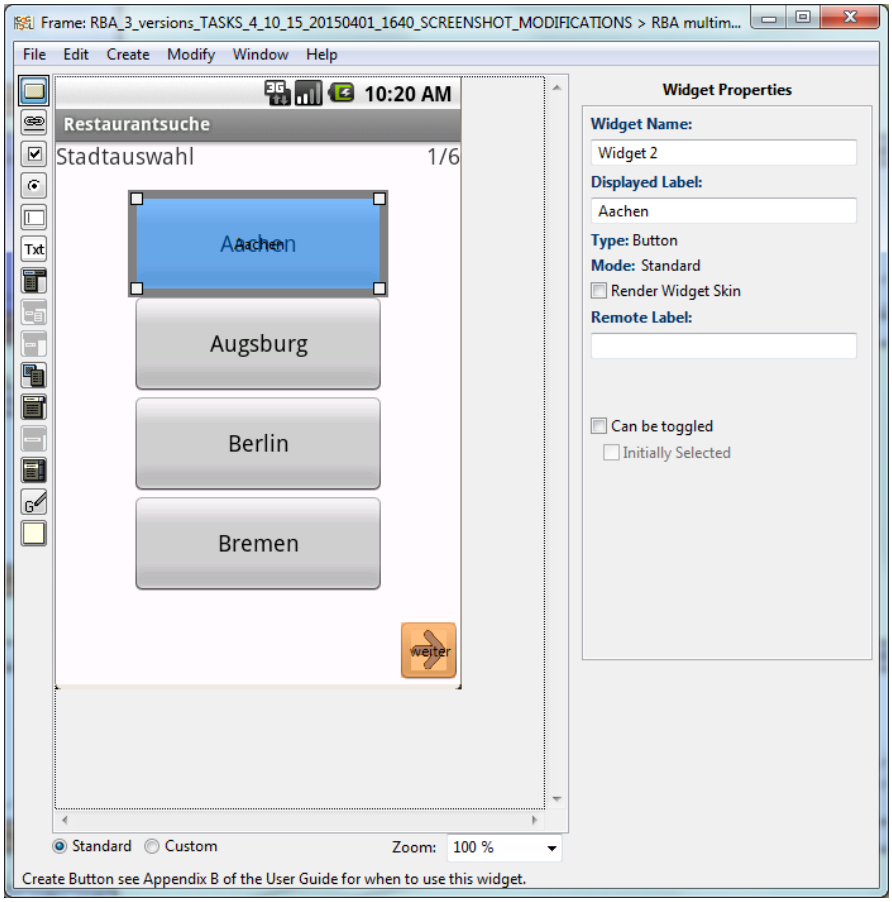

Fig. 5.11 The CogTool frame window for modeling elements of the graphical user interface.

Transitions between the frames are integrated in the design window. Figure 5.12 shows a detail of the design window. On the right side the properties of the RBA design can be viewed. The name of the design is ”RBA multimodal”. In Section 5.2.4 also unimodal designs are considered. The frames of the unimodal designs are the same as in this multimodal example. The designs only differ in the transitions. The RBA touch design only contains graphical transitions, and the RBA speech design only contains speech transitions. The RBA multimodal design contains both types of transitions in one design. All transitions are represented by black arrows. The graphical transitions connect buttons of a frame with followup frames. At the bottom of each frame a microphone is provided as a device for speech input. The speech input transitions connect the microphone of a frame with followup frames. Once a transition is entered the utterance triggering the transition must be specified.

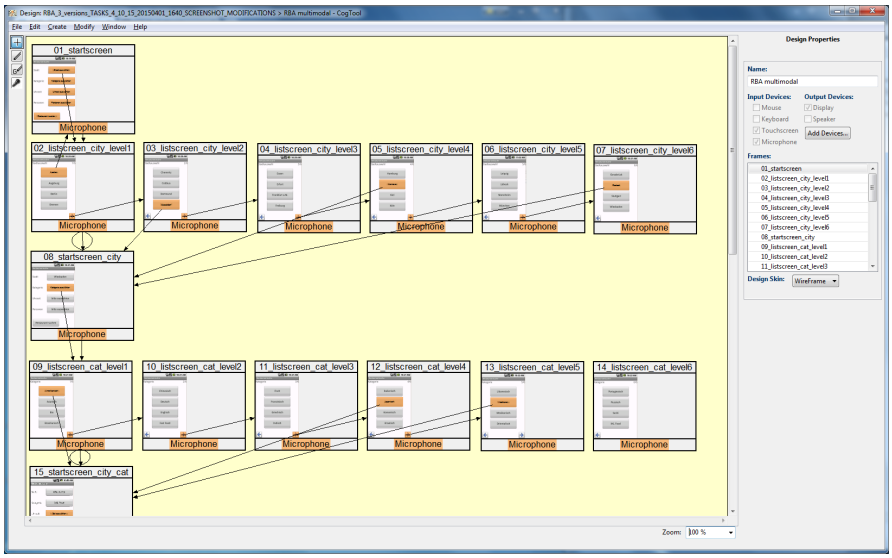

Fig. 5.12 A detail of the CogTool design window. Start screen and List screens are alternating line by line. All six list screens of a category are always in a single line.

In order to finalize a design the task of interest already has to be known by the modeler, as all necessary transitions between the frames of the design have to be modeled, so that the task can be demonstrated. Therefore the finished design of the RBA system includes all transitions for the tasks that are examined in this work. The tasks are further detailed in the next section.

5.2.2.2 Task Demonstration

In Section 5.2.3 the simulated modality selection behavior is analyzed. The analysis aims at examining the validity of the integration of the modality selection algorithm. Therefore it is sufficient to demonstrate a short task containing a modality selection for touch as well as for speech input and to merge the two unimodal solutions in the way described above. The variation of the input parameters of the modality selection algorithm can not be affected within CogTool. The ACT-R simulation also has no direct influence on the use of the algorithm. The parameters are so far not automatically determined but manually adjusted. To test the performance of the algorithm a single touch screen of the RBA design is sufficient. Other details of the examination are depicted in the respective section.

In Section 5.2.4 the total task duration of three specific tasks is predicted with two different CogTool models and one ACT-R model. Figure 5.13 shows the CogTool script window for demonstrating a task with the multimodal RBA design. On the left of the window the current frame is displayed. Tasks can be demonstrated by clicking on the interactive (orange) elements of the frame. A click on a button causes a transition to a subsequent frame. The current view of the script window shows a click on the microphone that opens a context menu as several utterances have been defined causing different transitions. By selecting one of these utterances the respective transition is triggered. On the right of the window the script generated by CogTool is displayed. By graphical or speech input in a frame steps are added to the script. The example shows the completed script for task 15 including the sub tasks: city=”Rostock”, culinary-category=”mediterran”, time=”elf uhr”, persons=”neunzehn personen”. The steps of the scripts are listed sequentially line by line from top to bottom. In the columns the used frame, the executed action, and the used device are displayed. The *Think-mod-sel* actions are manually integrated as modality selection steps before each input via a device. Once an input via a device is made CogTool automatically integrates another ”think” step into the script before the step for the actual input. In order to be able to better distinguish the ”think” steps, the names are changed. Each ”think” step for modality selection is followed by a modality specific ”think” step. In the current view line four including the second *Think-mod-sel* step is selected. The duration of the *Think-mod-sel* step is set to 0.1 seconds. The durations of the modality specific steps will be changed later in the ACT-R model. All the three tasks that were used for the performance prediction were demonstrated in the same way as depicted here for touch screen, for speech and for multimodal input. The CogTool script for touch screen input and for speech input provide the basis for the multimodal ACT-R models.

5.2.2.3 Human Performance Model

The ACT-R cognitive architecture is used with its default values. Regarding Fitts’ law the size and the position of the buttons are relevant for the performance prediction. The human performance model needs more time if buttons are smaller and if

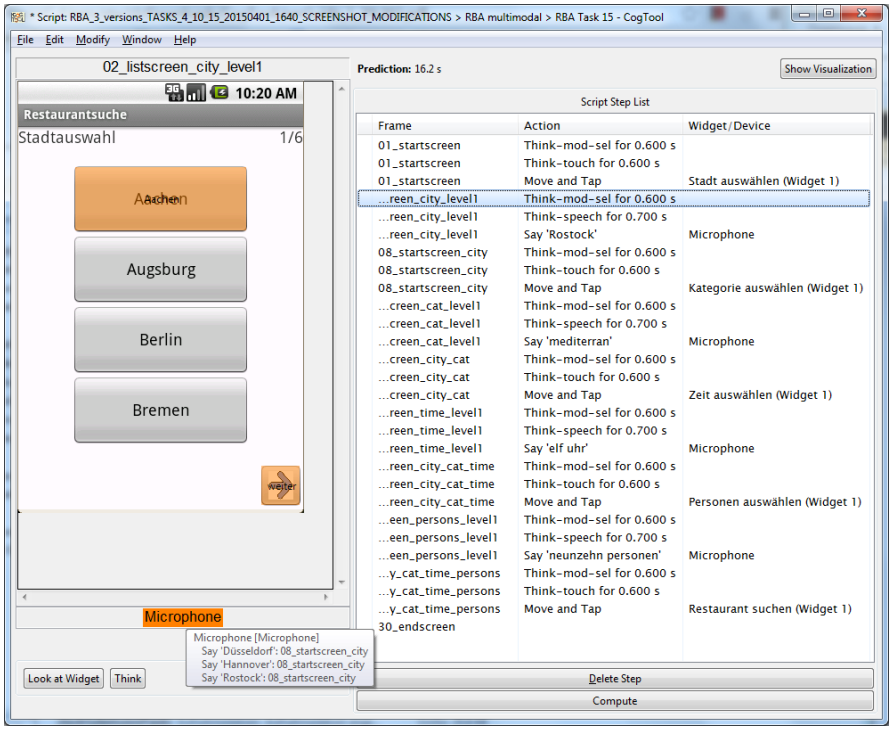

Fig. 5.13 A detail of the CogTool script window. It has to be noted that the durations of the shown ”think” steps are not the duration that were used for the model predictions.

the distance between the actual finger position and the button is greater. The buttons for selecting a list category on the start screen are placed one underneath the other and are same size and orientation. The button for sending the search request is same size, but is placed further on the left. On the list screens the buttons in the lists also are placed one underneath the other and are same size and orientation. The forward backward buttons are same size. The forward button is placed in the lower right corner and the backward button is placed in the lower left corner.

Regarding speech input the duration of an utterance depends on the number of characters. The human performance model uses 50 ms per character as an estimate for how long it takes the user to say something.

5.2.3 Analysis of Simulated Modality Selection Behavior

Similar to Section 5.1.3 in this section the integration of the modality selection model in ACT-R models generated by CogTool is validated. The baseline for the

analysis of the modality selection behavior of multimodal ACT-R models are therefore the predictions of the modality selection algorithm. These predictions serve as goal values for the modality selection algorithm that was integrated in the ACT-R models. Rendered multimodal an ACT-R model decides for a modality according to the probability calculated by the modality selection algorithm. As a result of the ACT-R simulations the predicted percentage of speech usage should be approximately the same as predicted by the algorithm. In order to validate the integration, in total 36 ACT-R simulations were performed: 6 simulations with different levels of modality efficiency for each of the 6 experimental conditions of input performance. Each simulation contains 128 iterations. As in the experiments with real humans interactions at the first list screen of the RBA were simulated. For each simulation log data for a specific level of modality efficiency and a specific error condition is generated. To adjust the error conditions and the levels of modality efficiency the input parameters of the modality selection function were adapted according to the experimental conditions.

The ACT-R trace of each simulation is stored in a separate log file. The trace of an ACT-R model contains the output of the simulation, including the actions and times of the perceptual-motor modules and the memory modules and other notifications. The log file of each task is imported into excel where the number of touch-screen and speech inputs were counted and the percentage of speech usage was calculated. Figure 5.14 illustrates the results of the simulations. For each error condition a diagram with the prediction results for each level of modality efficiency was produced.

As in the diagrams in the respective MeMo section the data predicted by the modality selection algorithm is colored dark gray. The runs of these baseline curves has already been discussed in Section 5.1.3.

The data simulated by the ACT-R model colored in light grey traces the curves of the modality selection algorithm quite well. The deviations are a result from the stochastic behavior of the modality selection algorithm. Speech is always selected with the calculated modality distribution. The sample of 128 simulation runs still shows in some cases that the total speech usage can be predicted a bit too high or too low. The overall fit is quite high with an R^2 of 0.99 and an *RMSE* of 0.04. The values for the single conditions are documented in Table 5.6.

Table 5.6 Performance measures of ACT-R predictions for the validation of the integration of the modality selection algorithm.

Condition	R^2	*RMSE*
$[T00, S00]$	0.995	0.040
$[T00, S10]$	0.994	0.047
$[T00, S20]$	0.996	0.045
$[T00, S30]$	0.998	0.054
$[T20, S20]$	0.984	0.034
$[T20, S00]$	0.991	0.018

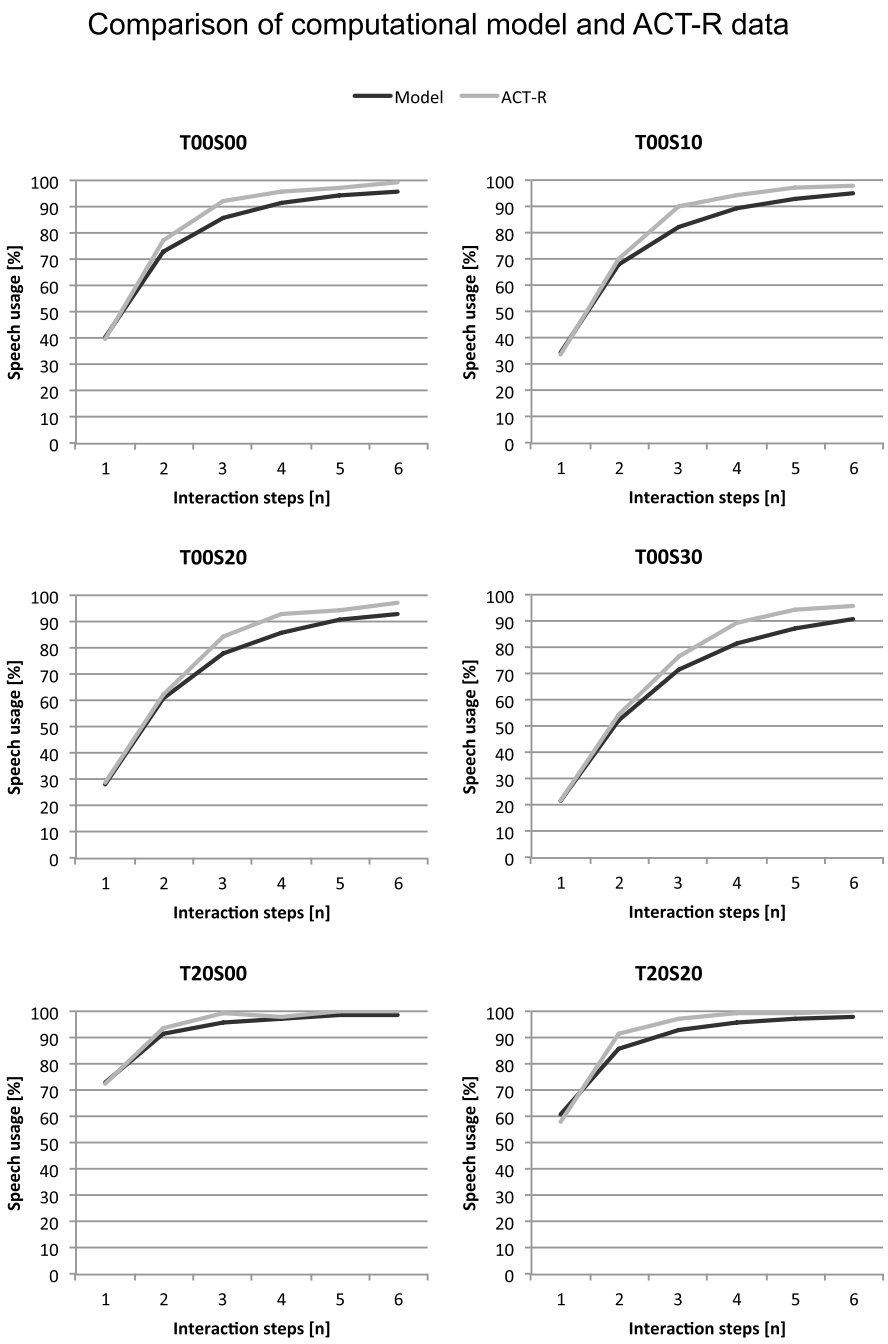

Fig. 5.14 Modality selection behavior of computational model and the CT/ACT-R simulation.

5.2.4 Application Example for the Prediction of Total Task Duration

The aim of this section is to test the usefulness of the multimodal ACT-R model generated as depicted in Section 5.2.1. The model contains the modality selection algorithm that has been developed in this work. As a baseline for the assessment unmodified CogTool predictions are considered. According to the stepwise procedure for generating the ACT-R code, at first the CogTool specifics of the application example are depicted. After that ACT-R specifics are described. The results of both CogTool and ACT-R predictions are finally compared to human data.

5.2.4.1 Specifics of the CogTool Conditions

The application for the prediction of total task duration is a classical example how CogTool can be used in an usability engineering context within a development stage where no real prototype is existing. A system designer working on the RBA may consider to integrate speech input, as speech shortcuts could be implemented in order to increase the efficiency of the application. At this stage comparable data of real humans is usually not available. Therefore CogTool predictions could be helpful in order to guide the designer in making decisions for the interaction design of the system. It would be useful to compare system designs for touch screen input, speech input, and multimodal input. Therefore RBA designs and tasks for this example are created with CogTool as depicted in Section 5.2.2. In order to establish a baseline for the predictions the task durations are calculated as defined by CogTool meaning that no additional "think" steps are integrated and that the duration of the automatically integrated "think" steps are not changed. This condition is called $CT_{default}$.

For another CogTool condition the modality specific durations for speech and touch screen input, obtained from the empirical data, are used. Additional "think" steps for modality selection are demonstrated and the durations are adjusted as in the ACT-R modification depicted in Section 5.2.1.2. Thus, the resulting CogTool script contains durations that are oriented on the human data. The analysis will also show how well these durations handle the differentiation between the reasoning process and the execution of speech or touch input. As this condition contains the modality specific durations for modality selection it is called CT_{ms}.

Both the condition $CT_{default}$ and the condition CT_{ms} are completely feasible with CogTool. According to the theory underlying CogTool condition $CT_{default}$ determines the processing time of a skilled user. Regarding modality selection the following interaction strategy was chosen: a skilled user always selects the modality that leads on with a smaller number of interaction steps; if this number is equal for touch screen and speech input, touch screen is selected, as touch screen input usually takes less time and is less error-prone. Thereby the selection of a modality requires no time for the skilled user.

In contrast, with respect to modality selection condition CT_{ms} uses other durations, obtained from the RBA experiments with real users. The subjects in the RBA experiments were no skilled users. However, the task to be performed was fairly easy. Particularly the modality selections may require decisions that are new or at least not yet practiced for most participants. In this work it is assumed that additional or adapted reasoning processes cause effort on the part of the users, resulting in longer durations for interaction steps including modality selections. The duration of a CogTool default "think" step is 1.2 seconds. As already described in Section 5.2.1.2 the experiments revealed for the RBA that touch screen input takes 0.8 seconds and speech input 2.0 seconds. Apart from the changes regarding the durations, the interaction strategy of the skilled user is utilized. Therefore for the predictions with CT_{ms} higher values for total task duration can be assumed at least for the tasks with high modality efficiency of speech input.

5.2.4.2 Specifics of the ACT-R Condition

The multimodal ACT-R model integrating the modality selection algorithm is based on the CogTool condition CT_{ms}. The CogTool scripts are generated by the touch and the speech versions of the RBA. All adaptions are made as depicted in Section 5.2.1.2. In these steps where the modality selection process has been integrated modalities are selected according to the specified input parameters. In the other steps originating from the CogTool where touch screen or speech input is performed nothing has been changed. By running the multimodal model several times task solutions differing in modality usage can be produced. For all these solutions an average task duration with a standard deviation can be calculated.

The ACT-R models are run 128 times. The ACT-R trace of each model is stored in a separate log file. The trace of an ACT-R model contains the output of the simulation, including the actions and times of the perceptual-motor modules and the memory modules and other notifications. The log file of each task is imported into excel where the total task durations are filtered and means and standard deviations are calculated.

5.2.4.3 Results

Figure 5.15 shows the results of the CogTool performance predictions for both created conditions, $CT_{default}$ and CT_{ms}. The touch screen based, the speech based, and the multimodal design are compared to each other.

It should be noted again that the modality efficiency of speech increases between the tasks. Figure 4.1 summarizes the assignment of speech benefits to tasks. The modality efficiency of speech steadily increases between the tasks except for a small exception in Task 10. For the touch screen design, a high speech benefit means that multiple touch screen interaction steps are required for the task. The higher the speech benefit, the more touch screen interaction steps are needed. Regarding

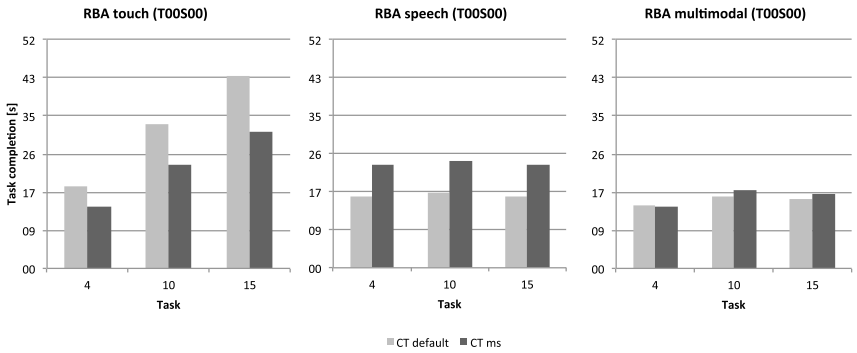

Fig. 5.15 Results of the CogTool predictions.

the CogTool predictions this increase of interaction steps causes almost linearly increasing total task completion times for both conditions. As shorter durations were assigned to the ”think” touch steps of the CogTool script of the CT_{ms} condition, also the total task completion times of all tasks are smaller. The difference becomes larger with an increasing number of interaction steps.

For the speech design the total task completion times for all tasks of both conditions are almost constant. Using speech all list elements are accessible on the first list screen, no matter on what list screen they are. As longer durations were assigned to the ”think” speech steps of the CogTool script of the CT_{ms} condition, also the total task completion times of all tasks are longer. Comparing the predictions of task 10 of the speech and the touch screen design condition $CT_{default}$ shows a smaller total task completion time for the speech design, while the prediction based on the of CT_{ms} is still slightly higher. For task 15 compared to the touch design the performance predictions of the speech design are better for both conditions.

Looking at the predictions for the multimodal design a slight increase of total task execution times can be observed between the tasks 4 and 10. In task 15 the total task execution time decreases slightly in both conditions. An explanation could be a specific difference between the tasks that was not considered when the tasks were constructed: taking all characters of the utterances for the selection of list elements together, task 10 has 50 characters in the CogTool script. Task 15, however, has only 40 characters. The difference in the predictions may result from the calculation of ACT-R that uses 50 ms as duration per character. In task 4 the modality usage pattern is different as touch screen is used for two sub tasks. The differences in the predictions for $CT_{default}$ and CT_{ms} are considerably lower. Compared to the touch screen and the speech design the performance could be improved for all tasks. The differences to the unimodal designs result from the strategy of the skilled user, always deciding for touch screen having a shorter duration if the modality efficiency of both modalities is equal in terms of interaction steps. If a speech shortcut in terms

of interaction steps is present speech input is used. The concrete values of all CogTool time predictions can be looked up in Appendix C.4.

Figure 5.16 shows the predictions of the multimodal ACT-R model in comparison to human data and both conditions of the multimodal CogTool design.

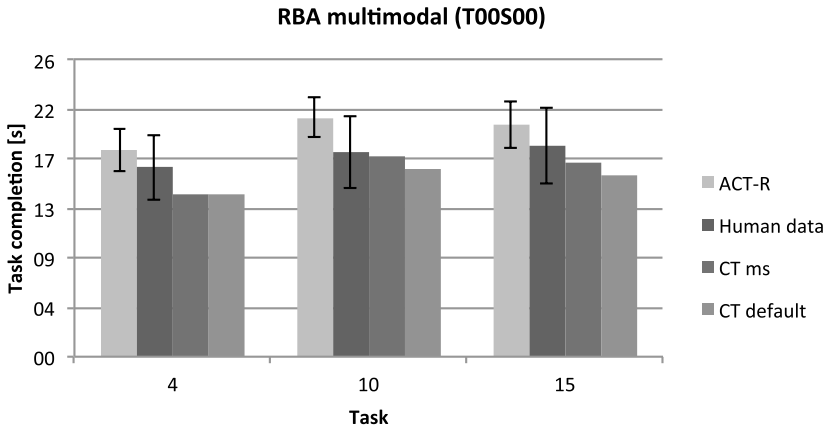

Fig. 5.16 Total task completion time comparison of empirical data and simulated data from ACT-R and CogTool in seconds for tasks 4, 10, and 15. Error bars show standard deviation.

The results of the MeMo simulation already revealed that different interaction strategies of real users can lead to outliers in the data, causing increased means and standard deviations. Although the study of individual differences of users is not in the focus of the experiments, the results show that these different interaction strategies exist among users. However, the results also show that the most other users follow the strategy pursued by the developed modality selection model. For the presentation of the human data in comparison to the ACT-R simulation results it was therefore decided to exclude data of one participant with a different interaction strategy, namely using touch screen only. The analysis of interaction times further revealed that two other participants appeared to have problems during task execution as they, had highly increased interaction times of 26.9 and 8.1 seconds in single task steps. As the average interaction time for one step is only 1.3 seconds (compare Section 5.2.1.2) these two were also excluded for the presentation[3]. Table 5.7 shows the corrected and the uncorrected values.

For the multimodal ACT-R model an increase of total task completion time can be observed between task 4 and task 10. In task 15, total task completion time slightly decreases compared to task 10. As explained above in this section, this

[3] For MeMo the following difference has to be considered: Regarding the MeMo simulation results high standard deviations of the model predictions give indications to interaction problems of the user model. Therefore it was decided to show the uncorrected data in the MeMo Figures.

Table 5.7 ACT-R Vs Human data Vs CogTool.

Data	Task	M	SD	M (corr)	SD (corr)
Human	4	16.4950	2.778		
ACT-R	4	18.000	1.583		
$CT_{default}$	4	14.2	-		
CT_{ms}	4	14.1	-		
Human	10	17.862	3.194		
ACT-R	10	20.908	1.708		
$CT_{default}$	10	16.4	-		
CT_{ms}	10	17.6	-		
Human	15	23.000	9.581	18.405	3.265
ACT-R	15	20.306	2.090		
$CT_{default}$	15	15.8	-		
CT_{ms}	15	17.0	-		

behavior was also observed in the CogTool predictions. The difference may again result from the way ACT-R calculates the duration of the utterances for speech input.

The standard deviations of the ACT-R simulation are consistently slightly smaller then the standard deviation in human data. The average total task completion time predicted with ACT-R is within the standard deviation of the human data for all tasks. Further the ACT-R averages are consistently higher then the averages of human data. At worst task 10 is predicted, where the average total task completion time predicted by ACT-R is about 3 seconds higher than the average of human data. Apparently the differentiation between modality specific ”think” step (also referred as the reasoning process) and the execution of speech or touch input is so far not optimally adjusted. A better fit to human data should be possible by parameter fitting for the respective durations.

It is correct that the CogTool predictions provide shorter total task completion times than the human data. CogTool predicts the performance of skilled users, however, the participants were not experts in using the system. Regarding the ACT-R simulation the durations of the modality specific ”think” steps and the ”think” modality selection step are adjusted to human data. As a consequence the ACT-R model does not predict the performance of skilled users according to the theory underlying CogTool. Only for the execution of user input the ACT-R model still uses the same mechanisms as CogTool. The CogTool predictions rather provide a lower limit for the ACT-R simulation. The tasks that were demonstrated by the modeler using CogTool can be seen as the best solution, commonly with the shortest possible total task execution time. During several iterations the ACT-R simulation also finds a best solution. However due to its stochastic behavior the multimodal simulation also finds other solutions with higher total task execution times. It is therefore correct that the ACT-R simulation provides a longer total task execution time than CogTool. The reason why the ACT-R predictions are consistently higher then the human data can be attributed to the above mentioned adjusted durations.

All in all, the results of performance prediction are plausible. The further discussion of the results will take place in Section 5.2.5.

5.2.4.4 CogTool Reports

Additionally to the results for the performance predictions, CogTool delivers a visualization of the ACT-R simulation (see also Section 2.4.5). The aim of this report is to gain insights into the efficiency of the designs and the activities of the single ACT-R modules. The CogTool report feature can only be utilized for tasks that were directly created with CogTool. It is not possible to import the results of the multimodal ACT-R simulation that was performed within the multimodal procedure. However the ACT-R runtime environment provides comparable features. The possibility of displaying and browsing through the activities of the ACT-R simulation provides details about the perception and the interactive behavior of the simulated user. For each frame of the design and for each ACT-R buffer details about starting time, duration, and end time can be viewed.

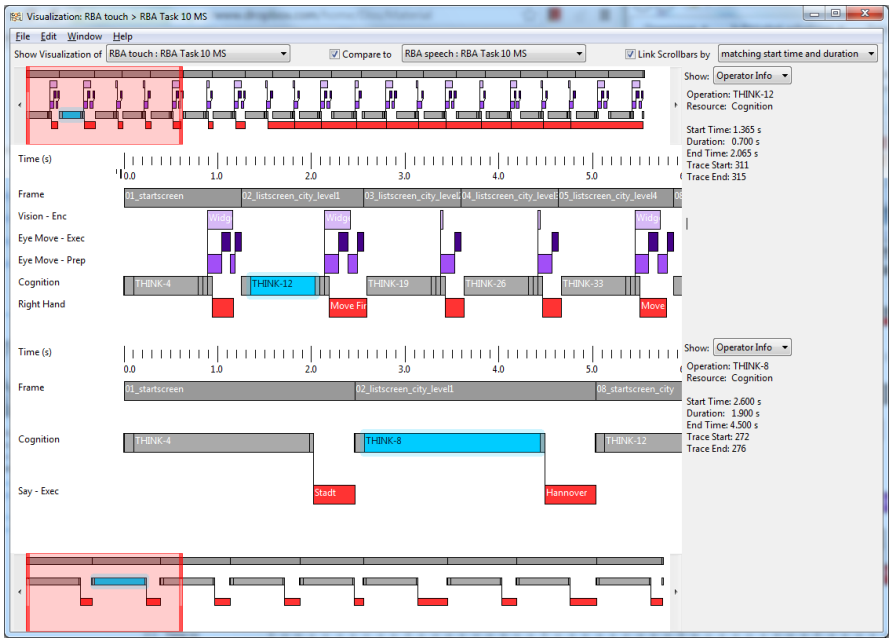

Fig. 5.17 The CogTool report window.

Figure 5.17 shows an example comparing two task demonstrations to each other. The task at the top is solved with the RBA touch screen design, while the task at the bottom is solved with the RBA speech design. At the top and at the bottom of

the report window an overview of the tasks is shown. The area marked in red in this overview represents the section that is shown below or above the overview. At the top of each section a timeline allows a rough chronological classification of the underlying actions of the cognitive architecture. In the first row under the timeline the frames of the design are represented. Comparing the first frame of two tasks it can be observed that the duration in the start screen is longer in the speech design. This difference can be mainly attributed to the different durations of the modality specific ”think” productions, which can be verified in the row named cognition. Below the cognition row in the upper task (with the touch screen design) the movement of the right hand is represented in red as an input action. The single actions in this line differ in duration. Here Fitts's law comes into play calculating the durations to select the elements in the UI depending on the size of the target and the distance between the target an the hand. Above the cognition row the visual encoding, and the execution and preparation of eye movements is represented. In the lower task (with the speech design) the input action is represented in the row named ”Say Exec”. In the example the utterance ”Stadt” is performed. The duration of the utterances is calculated by ACT-R. It is notable that the theory in this case does not simulate visual encoding and eye movement. After the input actions in the two different tasks the frame row changes to the frame *02_listscreen_city_level1* which represents the first list screen of the city list of the RBA. The sequence of the individual ACT-R actions is similar to the previous screen. It is notable that in the upper task the duration of the first and the second frame taken together are almost as long as the duration of first frame of the lower design. Again this can be attributed to the modality specific ”think” productions. After the second speech input in the second frame the simulated user gets back into the start frame, having selected the right city, while using touch screen the simulated user is still browsing through the list. Here the benefit of speech input, respectively the higher modality efficiency of speech takes effect.

5.2.5 Discussion

The multimodal procedure utilizing ACT-R models generated with CogTool in order to create multimodal ACT-R models works. The analysis of the simulated modality selection behavior of multimodal ACT-R models revealed that the modality selection algorithm is correctly implemented in those models.

The duration defined in the CogTool ”think” steps are a sticking point for the accuracy of the prediction results. However, the precise prediction of processing times is not a key issue in this work. It rather indicates how the modality selection algorithm can be employed. Therefore no parameter fitting was executed with respect to the durations. Through another fitting the durations may be strongly adapted to the RBS system. A generality of the results would thus not be given. It would be important to collect data with other systems to verify the external validity of the simulation. For the ACT-R simulation this would be important in two respects: firstly, the

external validity of the modality selection needs to be tested, secondly the modality specific durations should be substantiated by further empirical findings.

The multimodal procedure shows that simulations are possible with ACT-R models, created by CogTool, and being expanded to show details of multimodality. For the designer in particular the extension of the ACT-R models is a difficult and time consuming task. An extension of CogTool, directly integrating modality selection, should be possible with reasonable effort. The integration of a ”think” modality selection step could be implemented similar to the integration of a ”Look at step”. In contrast to the previous prediction, based on a demonstration of a touch screen solution and a demonstration of a speech input solution, a simulation with multiple iterations would be possible generating different solution paths with different interaction times.

Further research is needed for a realistic indication of times for reasoning processes. The results of the RBA experiments show that a time of 1.2 seconds, which is used by CogTool by default, may not fit for arbitrary modalities. For the process of modality selection other durations are possibly more appropriate. Whether a ”think” modality selection step should be provided with a duration, or whether all durations should be included in ”modality specific steps” would have to be clarified by further research. Thereby also the time for preparing an utterance for speech input has to be further investigated. By changing the durations of the ”think” modality specific steps also the standard deviation of human data could be better reproduced. In the next section a comparison between MeMo and CogTool is drawn.

5.3 Comparison of MeMo and CogTool Simulations

In this section MeMo and CogTool are compared to each other on the basis of the attributes described in Section 2.2.4 including the classification by means of a taxonomy, subjective assessment, and common performances measures. With regard to the classification and the subjective assessment, attributes suggested by Ivory and Hearst (2001) are adapted.

5.3.1 Classification of the Tools

5.3.1.1 Method Class and Type

Both, MeMo and CogTool, can be assigned to the method class simulation. Using MeMo user, task and system models to mimic a user interacting with an interface are utilized. The user model is based on the MHP (Card, 1981). The results of the interaction are multiple interaction paths through the system graph that can also be visually explored in the MeMo workbench. For further processing of the data the simulated activities can be exported into log files. As a special feature a PDF report can be exported including several quantitative measures. MeMo further integrates a HTML import functionality.

Using CogTool system designs and task demonstrations to mimic a user interacting with a interface are utilized. The user model employs a GOMS-like model realized with the cognitive architecture ACT-R (Anderson et al., 1997). The results of the interaction are performance predictions for skilled users. The generated ACT-R models can also be visually explored in CogTool. For further processing of the data the simulated activities can be exported into log files. The performance measures of multiple alternative designs with demonstrated tasks as well as the ACT-R models can be exported. CogTool further integrates a HTML import and export functionality.

5.3.1.2 Automation Type

Both, MeMo and CogTool, support the analysis of recorded data. By means of MeMo potential usability problems can be identified. During multiple iterations of a specific task different solutions for the task are found. The user model thereby can lose the optimal path. Missing error recovery strategies can then lead to outliers in the data uncovering interaction problems. The simulation outcome is dependent on the modeled system, task, and user group. The software so far can not automatically suggest improvements from the analysis. Multimodal simulations are automatically possible with MeMo.

By means of CogTool performance predictions for skilled users can be made. A modeler demonstrates a task and CogTool automatically generates a script that

is translated into the ACT-R cognitive architecture by pressing a button. The simulation outcome is dependent on the modeled design and task. The software only follows the demonstrated path and can so far not automatically suggest improvements. Multimodal simulations are so far not automatically possible with CogTool. In order to simulate multimodal interaction the multimodal procedure depicted in Section 5.2 has to be performed.

5.3.1.3 Effort Level

As mentioned in Section 2.2.4 the levels of effort suggested by the taxonomy are not necessarily ordered by the amount of effort required. The actual amount of effort depends on the method employed Ivory and Hearst (2001). Both, MeMo and CogTool, require the development of a system model and a task model. For both approaches the underlying user model can be used without special effort. However, the actual effort to create the models varies considerably. Especially using MeMo higher effort has to be spent for the creation of the system model. In order to enable the MeMo simulation information from the task model has to be transferred to the system model. Therefore it is necessary to explicitly specify the change of the values of single attribute value pairs of the system to perform a certain transition. The information must therefore be specified twice in the task knowledge and in the transitions of the system. This work must be done manually and is therefore time-consuming and error-prone.

CogTool in contrast, requires less effort to create the system design. After a task is demonstrated no further information must be specified in order to enable the performance predictions. However, applying the multimodal procedure also for CogTool the effort increases considerably. Besides the creation of the required basic ACT-R models with CogTool, these models must then be changed in Lisp in order to enable the multimodal simulation in the ACT-R runtime environment. These steps require a considerable amount of time and expertise.

5.3.2 Subjective Assessment

Regarding the RBA simulation with MeMo a usability problem was uncovered: when the user model accidentally entered a wrong list it was not able to carry on appropriately as the user interface does not contain a back button for leaving the list without selecting one of the list items. This should be changed before the real system is implemented. In so far effectiveness may be attributed to MeMo. Referring to the performance predictions effectiveness can also be attributed to CogTool and the multimodal ACT-R simulations. As it could be expected the CogTool predictions were below the human data and the ACT-R predictions above the CogTool data.

Ease of use can only be attributed to CogTool. Creating a design and demonstrating a task works relatively easy and fast. As already described for the effort level,

MeMo and the multimodal ACT-R simulation are not as easy to employ. However, the proper application of MeMo should be easy to learn, in comparison to the expertise which is necessary to adjust the Lisp-based ACT-R models.

Regarding applicability both tools are applicable for a wide range of systems. The applications ranges from spoken dialog systems over WIMP and Web UIs to smartphones applications, and others like remote controls.

5.3.3 Goodness of Fit

The results for the goodness of fit are extremely good with $R^2 = 0.98$ and $RMSE = 0.03$ for MeMo and $R^2 = 0.99$ and $RMSE = 0.04$ for CogTool and differ only slightly from each other. The reason for the good results and the small difference is that the tools were used for the prediction of data with which the modality selection algorithm was trained. This was done deliberately in order to test the correctness of the implementations and to find out how well the predictions of tools fit to the predictions of the bare algorithm.

5.3.4 Discussion

The classification by means of the taxonomy, the subjective assessment, and the goodness of fit measures reveal that MeMo and CogTool in combination with the multimodal ACT-R simulation, encounter on an equal footing in all disciplines. Method class and method type serve for the classification of the tools showing that the tools per se have different fields of application. MeMo finds different interactions paths through a system model and provides clues to potential usability problems, while CogTool predicts the total task execution times of skilled users.

The modeling of information in MeMo requires some practice. For simple tasks it would be desirable to transfer information such as UI button labels automatically as an information if a button is pressed. The button labels are often used in the task knowledge. Therefore modeling effort could be saved.

An important difference between the tools is, that in MeMo the selection of interactions during the simulation is based on probabilities. The actual interaction steps are automatically calculated depending on the task and interface description, instead of being pre-defined by the modeler as in CogTool. However, in CogTool no information like variables and consequences of transitions have to be defined. Nevertheless, the effort in MeMo can be worthwhile for evaluators who want to discover possible usability problems by simulation. MeMo automatically finds different solutions for the task while in CogTool each single solution has to be demonstrated.

The multimodal ACT-R simulation, based on models that were created by CogTool, requires a considerable amount of expertise. This additional effort could be

reduced if the necessary functionalities are directly integrated into CogTool, as proposed in Section 5.2.5.

The different specialization speaks against the fusion of the tools in a single tool. Instead, the compatibility of MeMo system models and CogTool system designs would be desirable. System models of MeMo could be imported into CogTool and used directly. For CogTool designs, conversely, the same should be possible. By means of the MeMo simulation interaction paths could be found, which are converted into CogTool scripts. This would allow to start from a more general user knowledge, which is defined in MeMo. Solutions generated by MeMo are translated into CogTool scripts enabling CogTool to predict the processing times.

The existing HTML import features represent a possible starting point for such a product chain. However, the suitability of HTML is questionable due to the degrees of freedom in the HTML creation. It might be better to define a different format to make the created models usable in both tools. Through the existing HTML export feature CogTool fits better in the software development process. Designs created with CogTool can thus be converted into first interactive prototypes that can be tested by real users. For both tools compatibility with modern interaction design tools would also be desirable.

5.4 Chapter Summary

This chapter exemplified the practical application of the modality selection algorithm developed in Chapter 4. The integration of the algorithm required several extensions under the hood of MeMo. The modality selection behavior showed good results. However, the whole task had to be split into smaller parts as memory issues arose during the simulation. The prediction of the number of interaction steps showed useful results. An interaction problem, namely the lack of a back button, could be detected. Certain user strategies, like using touch screen only, are not covered by the model, and can therefore cause deviations between human and model data.

Regarding the multimodal procedure utilizing CogTool in combination with additional ACT-R simulations the modality selection behavior showed also good results. The duration defined in the CogTool Think steps as well as the modality specific durations should be substantiated by further empirical findings. The extension of the ACT-R models is a difficult and time consuming task that could be overcome if the modality selection algorithm is integrated into the CogTool simulation. A realistic indication of times for reasoning processes of modality selection needs further research.

The tool comparison reveals that both tool have their strengths and weaknesses. The tools per se have different fields of application. If a comprehensive analysis based on AUE methods should be conducted both tools can be used in sequence. At first different interaction paths for a task could be found using MeMo, and after that the total task execution times of single path could be predicted using CogTool.

Chapter 6
Summary and Outlook

6.1 Summary

Future developments in HCI will enable sequential independent multimodal systems (SIMS), thereby enabling free choice of input modalities. Users' modality choice is moderated by various factors. The motivation of this work was to examine the factors of input performance and modality efficiency and to build a model enabling the prediction of modality usage. The usefulness of the model was demonstrated by the deployment in the AUE tools MeMo and CogTool.

The foundations of the work were laid in Chapter 2 starting with an introduction to multimodal interaction including theories of human decision making, and to the topic of automated usability evaluation. Further the two AUE tools MeMo and CogTool were introduced. Finally the research questions of the work were formulated.

In three experiments the empirical results reported in Chapter 3 reveal that users of multimodal systems adapt modality usage to estimated modality efficiency as well as to input performance of modalities. On the one hand, speech input is increasingly preferred if speech gets more efficient in terms of interaction steps. On the other hand, the usage probability of a modality decreases if its input performance is limited (e.g., due to ASR errors or touch screen malfunction). Previous research, mostly in line with these findings, revealed rather discrete insights into the continuum of parameters influencing modality choice (Bilici et al., 2000; Wechsung et al., 2010). The presented series of experiments describes the relationships of multiple factor levels and gives a coherent idea about essential moderators of modality choice. The empirical results turned out to be consistent across experiments. The empirical findings answer the first research question RQ1: significant effects of modality efficiency and input performance on modality selection in multimodal HCI can be disclosed by unified experimental investigations.

The presented theoretical foundations and the observed user behavior inspire a utility theory-driven model that is derived in Chapter 4. The model forecasts an average users' modality choice behavior with considerable predictive power. A model comparison revealed that an integrative model that incorporates data about all avail-

able input performance conditions is qualified for beneficial estimations of modality usage. Particularly high prediction performances on unseen data and on conditions resting on sparse data indicate reliability of the integrative model. If individual subject data is predicted, substantial variances in individual modality usage profiles lead to decreased accuracy. Individual users appear to have different interaction strategies than those demonstrated by the model. An application example showed that the model is able to simulate plausible interaction between an average user model and a system model. Predicted average speech usage is mostly in line with human data. The simulation was realized as a state machine, which is a common concept in the AUE area. The modality selection mechanism can therefore beneficially extend existing AUE tools. A utility-driven computational model of modality selection could be formed based on the empirical data, which provides an answer for research question 2 (RQ2).

The the utilization of the modality selection model for the MeMo workbench and for CogTool based simulations is presented in Chapter 5. The multimodal extension of the MeMo workbench was documented and the creation of a multimodal system model of the RBA was outlined. Compared to the predictions of the bare algorithm the modality selection behavior of the MeMo simulations showed good results. The corrected predictions of the total number of interaction steps of three tasks with different modality efficiency provide useful results for two different error conditions. In combination with the MeMo reports the prediction results provided valuable insights into the usability of multimodal interaction. The reports revealed realistic modality usage as well as different possible interaction strategies. The uncorrected prediction include interaction errors of the model indicating a usability problem of the RBA: within the list screens a back button is missing.

Regarding CogTool the development of a multimodal procedure is documented combining a touch screen ACT-R model and a speech ACT-R model generated by CogTool into one multimodal ACT-R model. The multimodal model is augmented with the modality selection algorithm. The modeling of the RBA with CogTool and the adaptions made to the multimodal ACT-R model are outlined. Also for the CogTool based predictions of modality selection the comparison to the predictions of the bare algorithm showed good results. The usefulness of the multimodal ACT-R model was illustrated by an application example for the prediction of total task duration. CogTool was used to generate baseline predictions that were compared to human data and the predictions of the multimodal ACT-R model. In combination with the CogTool reports the prediction results provided valuable insights into the usability of multimodal interaction. The results revealed realistic performance predictions. Further the ACT-R simulation provides several automatically generated task solutions which would have to be demonstrated manually with CogTool.

Both tools have been compared by means of a taxonomy, subjective assessment, and common performances measures. The results showed that both tools encounter on an equal footing in all disciplines and that the tools per se have different fields of application. It was concluded that the usage of both tools in sequence could be valuable if a comprehensive analysis based on AUE methods should be conducted.

The utilization of the derived model for modality selection in MeMo and CogTool answered research question 3 (RQ3).

6.2 Discussion and Future Work

For a designer of multimodal user interfaces AUE simulations provide working knowledge about the modality usage to be expected. As the model acceptably approximates the effects of modality efficiency and input performance, other factors of interests can be brought into the designers' attention. By means of simulation, variants of multimodal interfaces can easily be compared. Optimizing the multimodal interface design in very early stages of system development will save time and monetary costs, as design errors and usability issues can be addressed without user testing before a real prototype is available. Further interaction log data enables usability predictions for future multimodal systems. Relevant information for the typical system design question "(where/how) should speech input be integrated?" can be gathered.

Individual users appear to have different interaction strategies than those demonstrated by the model. The investigation of modality usage patterns could expose user groups that differ in interaction strategies. By deploying specialized models for these groups individual differences between users could be taken into account. Further, hedonic quality, context, and other factors like the ones described in Section 2.1.3 provider areas for the extension of the model. Concerning system errors it has to be noted that the error rate perceived by a system user can be very different from the real error rate and can further change over time. These factors were not considered in the studies outlined in this work and should therefore be part of future research.

So far the model lacks a rigorous explanation of how the benefit of speech usage is derived from the task information and the interface. Effects of subtasks on each other and the interplay between task and system should be part of future research.

The wide field of multimodal systems offers several possibilities for improving the model. An expansion to other modalities will be needed, since, for example, non-contact gesture or gaze interaction and other input methods will increasingly emerge in the future. From a technical point of view the model is able to deal with arbitrary modalities, as long as it can be assumed that perceived utility determines modality usage. Looking beyond the simplified list-browsing task, novel interaction techniques like flick gestures allowing quickly scrolling through lists have to be taken into account. Furthermore, the number of input modalities should be adjustable within the model. However, much more data will be needed to fit these special conditions.

The combination of modalities, demonstrated by Bolt's long established "put-that-there" paradigm (Bolt, 1980), should be considered. Combining modalities will disclose a vast number of new conditions for the model that are feasible if the necessary data for parameter fitting is available. A difficulty will be that interaction steps

are not easy to calculate for modern user interfaces integrating advanced interaction techniques.

In the field of automatic usability evaluation, typically the interaction with newly designed system models is simulated. The model's extrapolation performance to other systems has so far not been tested. The Restaurant Booking Application (RBA) was a prototypical use case and portability to other systems and tasks has to be demonstrated to disclose the validity of the model. Note as well that the user interface of the RBA is not the most efficient one. More efficient interfaces for both modalities could be provided by using different and optimized GUI components and a more natural speech interface. For improving the efficiency of interaction, multimodal systems often integrate mechanisms for multimodal error correction or context-specific ASR grammars, concepts so far not covered by the model. Furthermore, it should be possible to facilitate the model for other tasks allowing speech shortcuts, such as "keyword typing" or "speaking for searching" inside a database. The question arises if a data driven approach will scale in the future. One possibility to overcome this issue could be to investigate whether the utility-driven approach can be supported by insights referring to cognitive modeling and to other moderators of modality choice.

With respect to the used AUE tools MeMo and CogTool also a number of possibilities for improving the tools arise. A standardized import and export functionality would be valuable in order to enable a product chain making use of the strength of each single tool. Furthermore, the compatibility to existing interaction design tools and integrated development environments would be useful. By improving integration into the software development process, the acceptance and awareness of AUE could be increased.

References

Anderson, J. R., Matessa, M., and Lebiere, C. (1997). ACT-R: A Theory of Higher Level Cognition and Its Relation to Visual Attention. *Human-Computer Interaction*, 12:439–462.

Apple (2011). *Siri*. `http://www.forestry.ubc.ca/conservation/power/` [Accessed: 25 January 2013].

Balbo, S. (1995). Automatic evaluation of user interface usability: dream or reality. In *In Proceeding of the Queensland Computer-Human Interaction Symposium. QCHI95*.

Bellamy, R., John, B., and Kogan, S. (2011). Deploying cogtool: Integrating quantitative usability assessment into real-world software development. In *Software Engineering (ICSE), 2011 33rd International Conference on*, pages 691–700. IEEE.

Bernsen, N. O. (2008). Multimodality theory. In *Multimodal User Interfaces*, pages 5–29. Springer.

Beuter, N. (2007). Gestenbasierte Positionsreferenzierung für die multimodale Interaktion mit einem anthropomorphen Robotersystem. Master's thesis, Faculty of Technology: Bielefeld University.

Bevan, N. (1995). Usability is quality of use. *Advances in Human Factors/Ergonomics*, 20:349–354.

Bilici, V., Krahmer, E., te Riele, S., and Veldhuis, R. N. (2000). Preferred modalities in dialogue systems. In *INTERSPEECH*, pages 727–730.

Bohn, J., Coroama, V., Langheinrich, M., Mattern, F., and Rohs, M. (2005). Social, economic, and ethical implications of ambient intelligence and ubiquitous computing. *Ambient intelligence*, pages 5–29.

Bolt, R. A. (1980). "Put-that-there": Voice and gesture at the graphics interface. In *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '80, pages 262–270, New York, NY, USA. ACM.

Card, S., Mackinlay, J., and Robertson, G. (1990). The design space of input devices. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people*, pages 117–124. ACM.

Card, S. K. (1981). The model human processor: A model for making engineering calculations of human performance. In *Proceedings of the Human Factors and*

Ergonomics Society Annual Meeting, volume 25, pages 301–305. SAGE Publications.

Card, S. K., Moran, T. P., and Newell, A. (1980). The keystroke-level model for user performance time with interactive systems. *Communications of the ACM*, 23(7):396–410.

Charwat, H. J. (1992). *Lexikon der Mensch-Maschine-Kommunikation*. Oldenbourg.

Dey, A. K. and Häkkilä, J. (2008). Context-awareness and mobile devices. *User interface design and evaluation for mobile technology*, 1:205–217.

Duffy, L. (1993). Team decision-making biases: An information-processing perspective. In Klein, G. A., Orasanu, J., and Calderwood, R., editors, *Decision making in action: Models and methods*, pages 346–359. Ablex Publishing, Norwood, NJ.

Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):32–64.

Engelbrecht, K.-P., Kruppa, M., Möller, S., and Quade, M. (2008). MeMo workbench for semi-automated usability testing. In *Proceedings of the 9th Annual Conference of the International Speech Communication Association (Interspeech '08)*, pages 1662–1665, Brisbane, Queensland, Australia.

Engesser, H. and Claus, V. (1993). *Duden "Informatik": ein Sachlexikon für Studium und Praxis*. Dudenverl.

Fiscus, J., Ajot, J., and Garofolo, J. (2008). The rich transcription 2007 meeting recognition evaluation. *Multimodal Technologies for Perception of Humans*, pages 373–389.

Franz, A., Henzinger, M., Brin, S., and Milch, B. (2006). Voice interface for a search engine. US Patent 7,027,987.

Fu, W. and Gray, W. (2006). Suboptimal tradeoffs in information seeking. *Cognitive Psychology*, 52(3):195–242.

Gibbon, D., Mertins, I., and Moore, R. K. (2000). *Handbook of Multimodal and Spoken Dialogue Systems: Resources, Terminology and Product Evaluation*. Kluwer Academic Publishers, Norwell, MA, USA.

Gray, W., Sims, C., Fu, W., and Schoelles, M. (2006). The soft constraints hypothesis: a rational analysis approach to resource allocation for interactive behavior. *Psychological review*, 113(3):461.

Hassenzahl, M., Burmester, M., and Koller, F. (2003). AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität. *Mensch & Computer 2003: Interaktion in Bewegung*.

Hedicke, V. (2000). Multimodalität in Mensch-Maschine-Schnittstellen. *Mensch-Maschine-Systemtechnik*, 2:203–232.

Hone, K. S. and Graham, R. (2000). Towards a tool for the subjective assessment of speech system interfaces (sassi). *Natural Language Engineering*, 6:287–305.

ISO9241-11 (1998). ISO 9241-11: Ergonomic requirements for office work with visual display terminals (VDTs) - part 11: Guidance on usability. Technical report, International Organization for Standardization (ISO), Geneva.

ISO9241-210 (2009). ISO 9241-210: Ergonomics of human system interaction-part 210: Human-centred design for interactive systems. Technical report, International Organization for Standardization (ISO), Geneva.

Ivory, M. Y. and Hearst, M. A. (2001). The state of the art in automating usability evaluation of user interfaces. *ACM Transactions on Computer-Human Interaction*, 33(4):470–516.

Iwata, H., Yano, H., Uemura, T., and Moriya, T. (2004). Food simulator: A haptic interface for biting. In *Virtual Reality, 2004. Proceedings. IEEE*, pages 51–57. IEEE.

Jameson, A., Mahr, A., Kruppa, M., Rieger, A., and Schleicher, R. (2007). Looking for unexpected consequences of interface design decisions: The memo workbench. In Winckler, M., Johnson, H., and Palanque, P. A., editors, *Proceedings of the 6th International workshop on TAsk MOdels and DIAgrams (TAMODIA)*, volume 4849 of *Lecture Notes in Computer Science*, pages 279–286, Toulouse, France. Springer.

John, B. (2012). CogTool User Guide. Technical report, IBM T. J. Watson Research Center, Software Productivity, 19 Skyline Drive, Hawthorne NY 10532.

John, B. E. and Kieras, D. E. (1996). The goms family of user interface analysis techniques: comparison and contrast. *ACM Transactions on Computer-Human Interaction*, 3:320–351.

John, B. E., Prevas, K., Salvucci, D. D., and Koedinger, K. (2004). Predictive human performance modeling made easy. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '04, pages 455–462, New York, NY, USA. ACM.

Jokinen, K. and Cheng, F. (2010). *New Trends in Speech-based Interactive Systems*. Springer Publishers.

Jordan, P. W. (2002). *Designing pleasurable products: An introduction to the new human factors*. CRC press.

Jungermann, H., Pfister, H.-R., and Fischer, K. (1998). *Die Psychologie der Entscheidung*. Heidelberg: Spektrum.

Kahneman, D. (2011). *Thinking, fast and slow*. Farrar, Straus and Giroux.

Kieras, D. E., Wood, S. D., and Meyer, D. E. (1997). Predictive engineering models based on the epic architecture for a multimodal high-performance human-computer interaction task. *ACM Trans. Comput.-Hum. Interact.*, 4(3):230–275.

Kühnel, C., Westermann, T., Weiss, B., and Möller, S. (2010). Evaluating multimodal systems: a comparison of established questionnaires and interaction parameters. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, pages 286–294. ACM.

Lewis, C. and Mack, R. (1982). Learning to use a text processing system: Evidence from thinking aloud protocols. In *Proceedings of the 1982 conference on Human factors in computing systems*, pages 387–392. ACM.

Mahlke, S. and Minge, M. (2008). Consideration of multiple components of emotions in human-technology interaction. In *Affect and emotion in human-computer interaction*, pages 51–62. Springer.

McCracken, J. and Aldrich, T. (1984). Analyses of selected lhx mission functions: Implications for operator workload and system automation goals. Technical report, DTIC Document.

Metze, F., Wechsung, I., Schaffer, S., Seebode, J., and Möller, S. (2009). Reliable evaluation of multimodal dialogue systems. In *Proceedings of the 13th International Conference on Human-Computer Interaction (HCII '09)*, pages 75–83, San Diego, CA, USA. HCII 2009.

Möller, S., Engelbrecht, K.-P., Kühnel, C., Wechsung, I., and Weiss, B. (2009). A taxonomy of quality of service and quality of experience of multimodal human-machine interaction. In *Quality of Multimedia Experience, 2009. QoMEx 2009. International Workshop on*, pages 7–12. IEEE.

Möller, S., Englert, R., Engelbrecht, K., Hafner, V., Jameson, A., Oulasvirta, A., Raake, A., and Reithinger, N. (2006). Memo: Towards automatic usability evaluation of spoken dialogue services by user error simulations. In *Proceedings of the 9th International Conference on Spoken Language Processing (Interspeech '06)*, pages 1786–1789, Pittsburgh, PA, USA.

Möller, S., Kühnel, C., and Weiss, B. (2011). Parameters describing the interaction with multimodal dialogue systems. ITU-T Recommendation Supplement 25 to P-Series, International Telecommunication Union, Geneva, Switzerland.

Morrison, J. (2003). A review of computer-based human behavior representations and their relation to military simulations. Technical report, DTIC Document.

Muthig, K.-P. (1999). Kognitive Prozesse: Aufnahme und Verarbeitung von Informationen. *Arbeits-und Organisationspsychologie, Weinheim*, pages 251–278.

Naumann, A. B., Wechsung, I., and Hurtienne, J. (2009). Multimodal interaction: Intuitive, robust, and preferred? In *Proceedings of the 12th IFIP TC 13 International Conference (INTERACT 2009)*, pages 93–96, Uppsala, Sweden. Springer.

Naumann, A. B., Wechsung, I., and Möller, S. (2008). Factors influencing modality choice in multimodal applications. In André, E., Dybkjæ r, L., Minker, W., Neumann, H., Pieraccini, R., and Weber, M., editors, *Perception in Multimodal Dialogue Systems*, volume 5078 of *Lecture Notes in Artificial Intelligence*, pages 37–43. Springer, Heidelberg, Germany.

Nickel, P., Eilers, K., Seehase, L., and Nachreiner, F. (2002). Zur Reliabilität, Validität, Sensivität und Diagnostizität von Herzfrequenz-und Herzfrequenzvariabilitätsmaßen als Indikatoren psychischer Belastung. *Z.Arb.Wiss*.

Nielsen, J. (1993). *Usability Engineering*, chapter 5, pages 115–163. Morgan Kau.

Nielsen, J. (1994). Heuristic evaluation. In Nielsen, J. and Mack, R. L., editors, *Usability Inspection Methods*, pages 25–62. Wiley and Sons.

Nigay, J. C. L., Jambon, F., and Coutaz, J. (1995). Formal specification of multimodality. In *CHI95 Workshop*.

Nigay, L. and Coutaz, J. (1993). A design space for multimodal systems: concurrent processing and data fusion. In *Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*, CHI '93, pages 172–178, New York, NY, USA. ACM.

Norman, D. A. (1990). The'problem'with automation: inappropriate feedback and interaction, not'over-automation'. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 327(1241):585–593.

Nurminen, A., Sirvio, K., Schaffer, S., Marconi, A., and Valetto, G. (2015). End-user applications techniques and tools. Technical report, FP7-SMARTCITIES-2013, EU Project STREETLIFE.

Obrenovic, Z., Abascal, J., and Starcevic, D. (2007). Universal accessibility as a multimodal design issue. *Communications of the ACM*, 50(5):83–88.

Oviatt, S. (2003). Multimodal interfaces. *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications*, pages 286–304.

Perakakis, M. and Potamianos, A. (2008). Multimodal system evaluation using modality efficiency and synergy metrics. In *Proceedings of the 10th international conference on Multimodal interfaces*, pages 9–16. ACM.

Potamianos, A. and Perakakis, M. (2008). Design principles for multimodal spoken dialogue systems. In Maragos, P., Potamianos, A., and Gros, P., editors, *Multimodal Processing and Interaction*, volume 33 of *Multimedia Systems and Applications*, pages 1–18. Springer US.

Raisamo, R. (1999). *Multimodal Human-Computer Interaction: a constructive and empirical study*. Tampereen yliopisto.

Rauterberg, M. (1996). A petri net based analyzing and modeling tool kit for logfiles in humancomputer interaction. *Proceedings of Cognitive Systems Engineering in Process Control*, pages 268–275.

Reder, L. M. (1988). Strategic control of retrieval strategies. *The psychology of learning and motivation*, 22:227–259.

Reisig, W. and Rozenberg, G. (1998). *Lectures on Petri Nets I: Basic Models: Advances in Petri Nets*, volume 1. Springer.

Ren, X., Zhang, G., and Dai, G. (2000). An experimental study of input modes for multimodal human-computer interaction. In Tan, T., Shi, Y., and Gao, W., editors, *Advances in Multimodal Interfaces ICMI 2000*, volume 1948 of *Lecture Notes in Computer Science*, pages 49–56. Springer Berlin Heidelberg.

Rudnicky, A. (1993). Mode preference in a simple data-retrieval task. In *Proceedings of the workshop on Human Language Technology*, pages 364–369. Association for Computational Linguistics.

Salvucci, D. D. (2009). Rapid prototyping and evaluation of in-vehicle interfaces. *ACM Transactions on Computer-Human Interaction*, 16(2):9–33.

Schaffer, S., Jöckel, B., Wechsung, I., Schleicher, R., and Möller, S. (2011a). Modality selection and perceived mental effort in a mobile application. In *Proc. 12th Ann. Conf. of the Int. Speech Communication Assoc. (Interspeech 2011)*, pages 2253–2256, Florence, Italy. International Speech Communication Association (ISCA).

Schaffer, S. and Minge, M. (2012). Error-prone voice and graphical user interfaces in a mobile application. In *Sprachkommunikation: Beiträge zur 10. ITG-Fachtagung vom 26. bis 28. September 2012 in Braunschweig*, pages 1–4. VDE-Verlag.

Schaffer, S. and Reitter, D. (2012). Modeling efficiency-guided modality choice in voice and graphical user interfaces. In Rußwinkel, N. and Drewitz, U., editors, *11th International Conference on Cognitive Modeling (ICCM2012)*, pages 253–254. Universitätsverlag der TU Berlin.

Schaffer, S., Ruß, A., and Reithinger, N. (2016). User guided speech technology integration for a mobility application. In *submitted to CHI '16*, San Jose, CA, USA.

Schaffer, S., Schleicher, R., and Möller, S. (2011b). Measuring cognitive load for different input modalities. In *9. Berliner Werkstatt Mensch-Maschine-Systeme*, volume Fortschritt-Berichte VDI of *22*, pages 287–292, Berlin, Germany. VDI Verlag.

Schaffer, S., Schleicher, R., and Möller, S. (2015). Modeling input modality choice in mobile graphical and speech interfaces. *International Journal of Human-Computer Studies*, 75:21–34.

Schleicher, R. and Wechsung, I. (2012). Modelling modality choice using task parameters and perceived quality. In *Sprachkommunikation: Beiträge zur 10. ITG-Fachtagung vom 26. bis 28. September 2012 in Braunschweig*. VDE-Verlag.

Schulz, M. (2014). *Simulation des Interaktionsverhaltens von Senioren bei der Benutzung von mobilen EndgerŁten*. PhD thesis, Springer.

Sheridan, T. and Parasuraman, R. (2000). Human versus automation in responding to failures: An expected-value analysis. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 42(3):403–407.

Streitz, N. A. (1988). *Psychologische Aspekte der Mensch-Computer-Interaktion*. Gesellschaft fur Mathematik und Datenverarbeitung.

Suhm, B., Myers, B., and Waibel, A. (1999). Model-based and empirical evaluation of multimodal interactive error correction. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, pages 584–591. ACM.

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive science*, 12(2):257–285.

Thüring, M. (2002). Kognitionspsychologische Prinzipien des Designs Grafischer Benutzungsoberflächen für Hypermediasysteme. In Marzi, R., Karavezyris, V., and Timpe, K., editors, *Bedienen und Verstehen. 4. Berliner Werkstatt Mensch-Maschine-Systeme*, pages 27–41, Düsseldorf. VDI-Verlag.

Timpe, K.-P. and Kolrep, H. (2002). Das Mensch-Maschine-System als interdisziplinärer Gegenstand. *Mensch-Maschine-Systemtechnik*, 2:9–40.

Turunen, M., Hakulinen, J., and Heimonen, T. (2010). Assessment of spoken and multimodal applications: Lessons learned from laboratory and field studies. In *Eleventh Annual Conference of the International Speech Communication Association*.

Tversky, A. and Kahneman, D. (1992). Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and uncertainty*, 5(4):297–323.

Tversky, A., Kahneman, D., and Moser, P. (1990). Judgment under uncertainty: Heuristics and biases. *Rationality in action: Contemporary approaches*, pages 171–188.

Varga, I., Aalburg, S., Andrassy, B., Astrov, S., Bauer, J., Beaugeant, C., Geißler, C., and Hoge, H. (2002). ASR in mobile phones-an industrial approach. *Speech and Audio Processing, IEEE Transactions on*, 10(8):562–569.

Wahlster, W. (2003). Towards symmetric multimodality: Fusion and fission of speech, gesture, and facial expression. In *KI 2003: Advances in Artificial Intelligence*, pages 1–18. Springer.

Wechsung, I. (2014). *An evaluation framework for multimodal interaction*. PhD thesis, Springer.

Wechsung, I., Engelbrecht, K., Naumann, A., Möller, S., Schaffer, S., and Schleicher, R. (2010). Investigating modality selection strategies. In *Spoken Language Technology Workshop (SLT), 2010 IEEE*, pages 31–36. IEEE.

Wechsung, I., Engelbrecht, K.-P., Kühnel, C., Möller, S., and Weiss, B. (2012). Measuring the quality of service and quality of experience of multimodal human–machine interaction. *Journal on Multimodal User Interfaces*, 6(1-2):73–85.

Weidenmann, B. (1995). Multicodierung und multimodalität im lernprozess. *Informationen und Lernen mit Multimedia, Weinheim: Psychologische Verlagsunion*.

Wharton, C., Rieman, J., Lewis, C., and Polson, P. (1994). The cognitive walkthrough method: A practioner's guide. In *Usability Inspection Methods*, chapter 5, pages 105–140. Wiley and Sons.

Wickens, C. D. (2002). Multiple resources and performance prediction. *Theoretical Issues in Ergonomics Science*, 3:159–177.

Wickens, C. D. and Hollands, J. G. (2000). *Engineering Psychology and Human Performance*. Upper Saddle River, EUA : Prentice-Hall.

Wolpaw, J. and Wolpaw, E. W. (2012). *Brain-computer interfaces: principles and practice*. Oxford University Press.

Zsambok, C. E. and Klein, G. (2014). *Naturalistic decision making*. Psychology Press.

Appendix A
System and Experiments Details

A.1 Start screen of the RBA

Fig. A.1 Start screen with no list items selected.

A.2 City list screens of the RBA

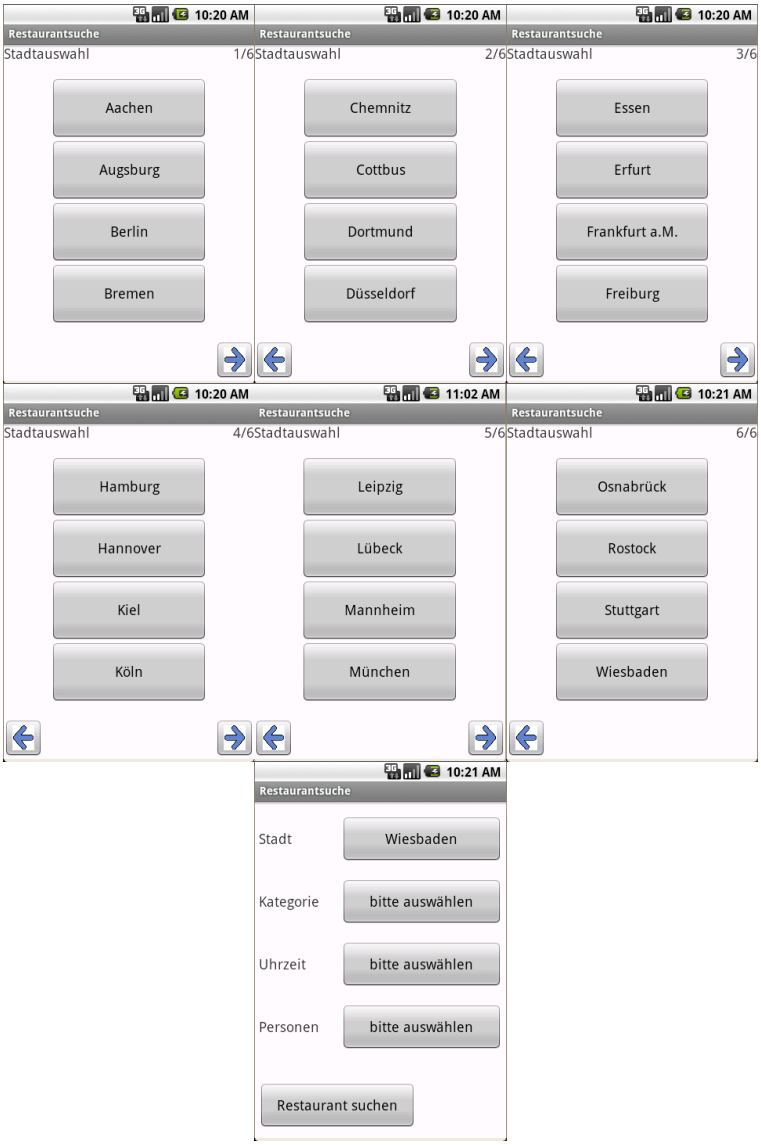

Fig. A.2 City list screens and start screen with the selected item ”Wiesbaden”.

A.3 Category list screens of the RBA

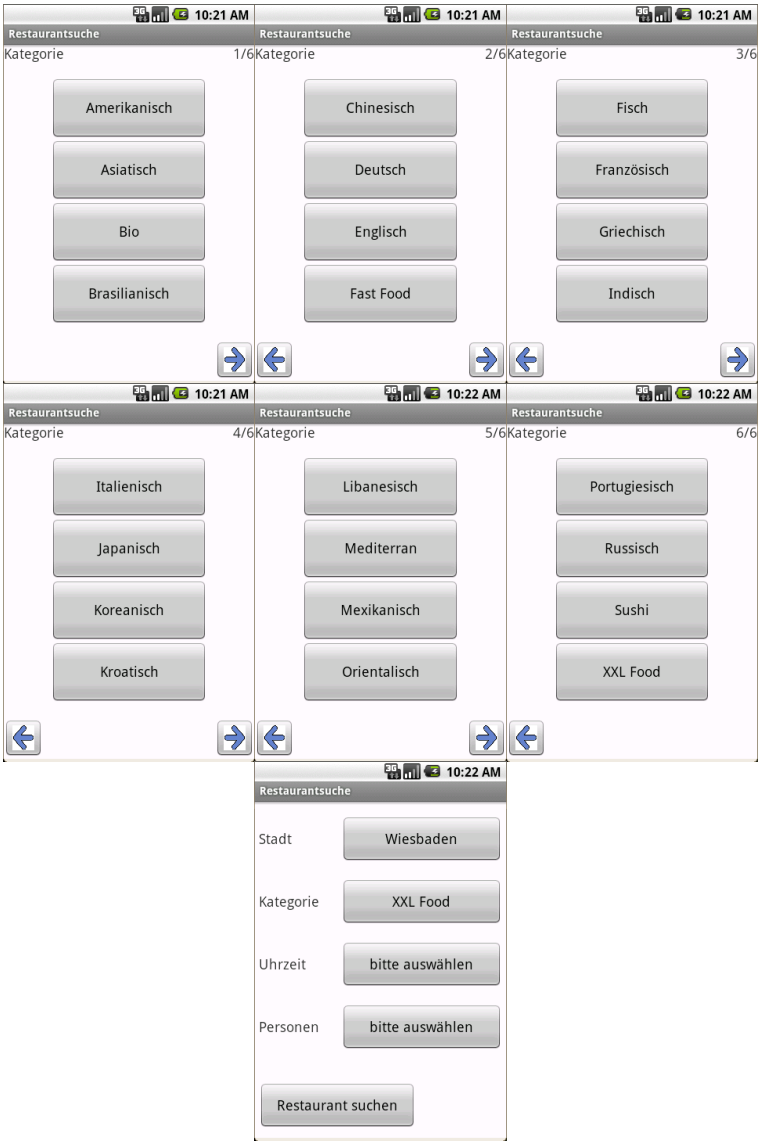

Fig. A.3 Category list screens and start screen with the additionally selected item ”XXL Food”.

A.4 Time list screens of the RBA

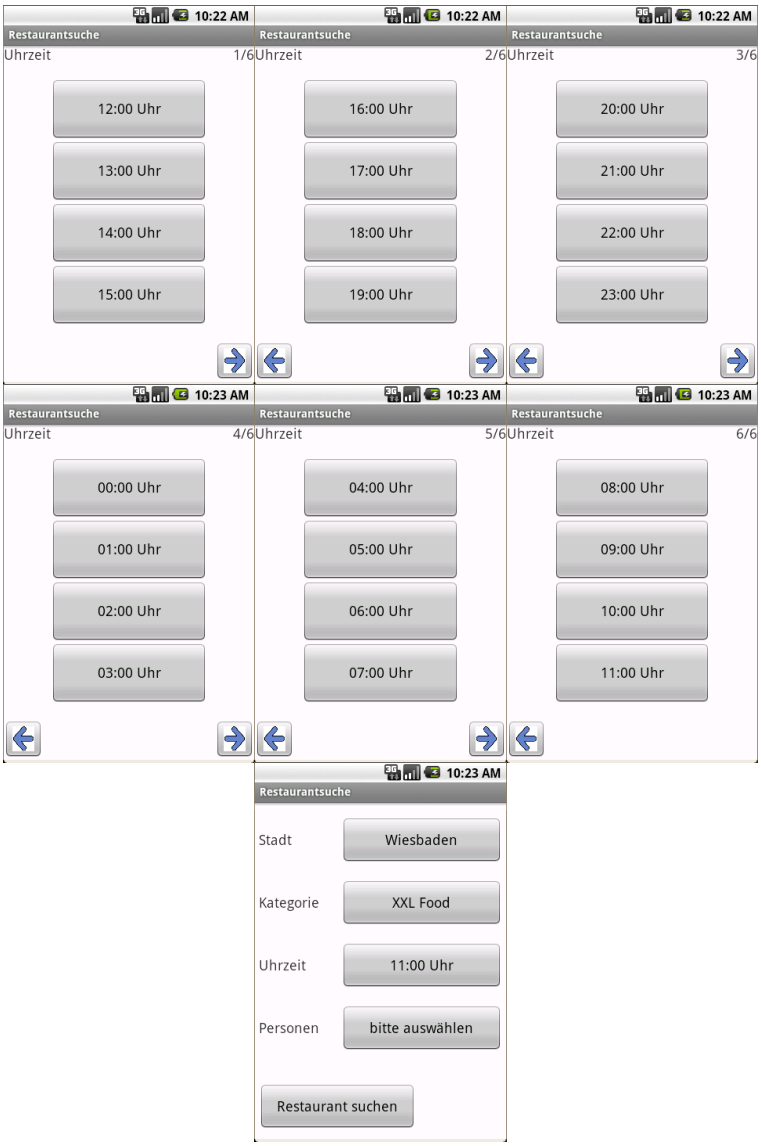

Fig. A.4 Time list screens and start screen with the additionally selected item ”11:00 Uhr”.

A.5 Persons list screens of the RBA

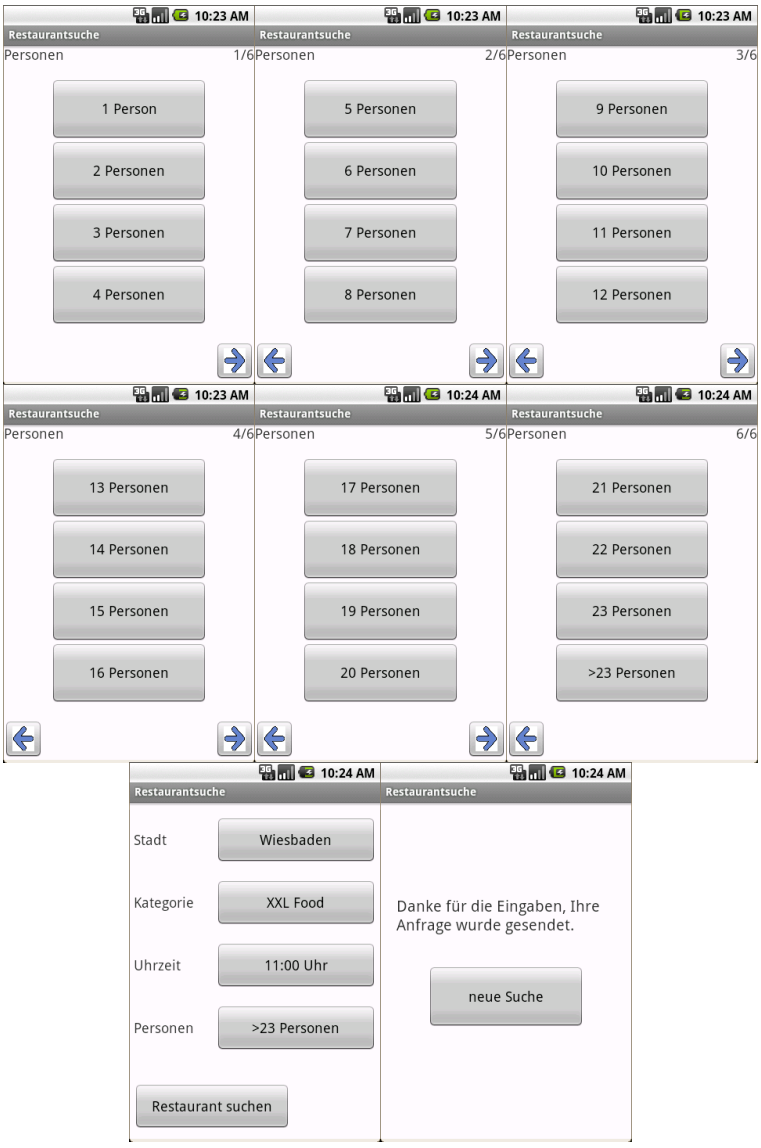

Fig. A.5 Persons list screens and start screen with the additionally selected item ”¿23 Personen”.

A.6 ASR Error GUI Feedback

Fig. A.6 The GUI feedback if an ASR error occurs.

A.7 Wizard-of-Oz Interface

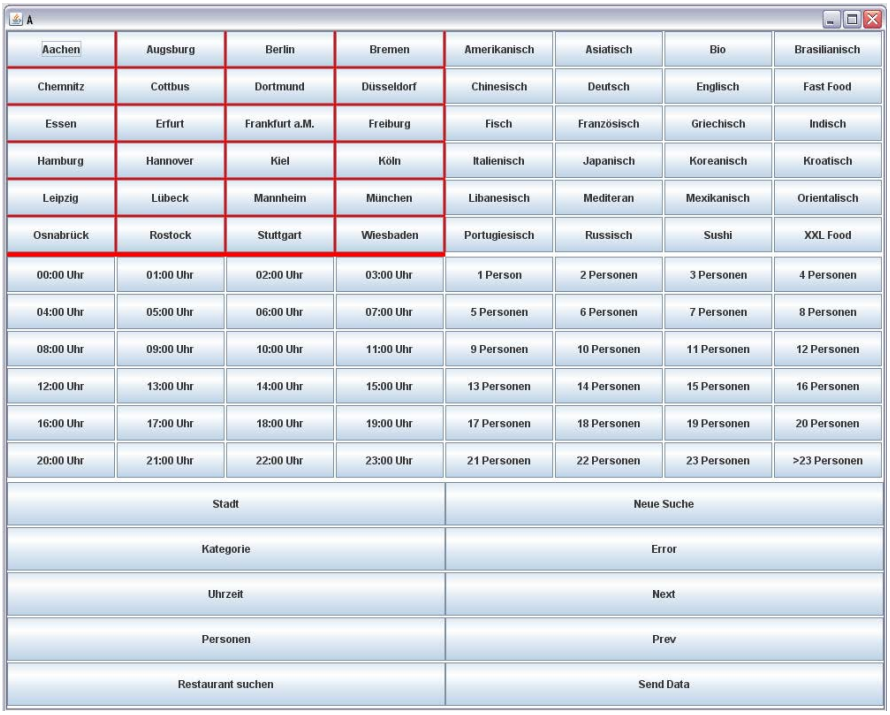

Fig. A.7 The Wizard-of-Oz WIMP interface.

A.8 Setup of the Experimentes 1 and 2

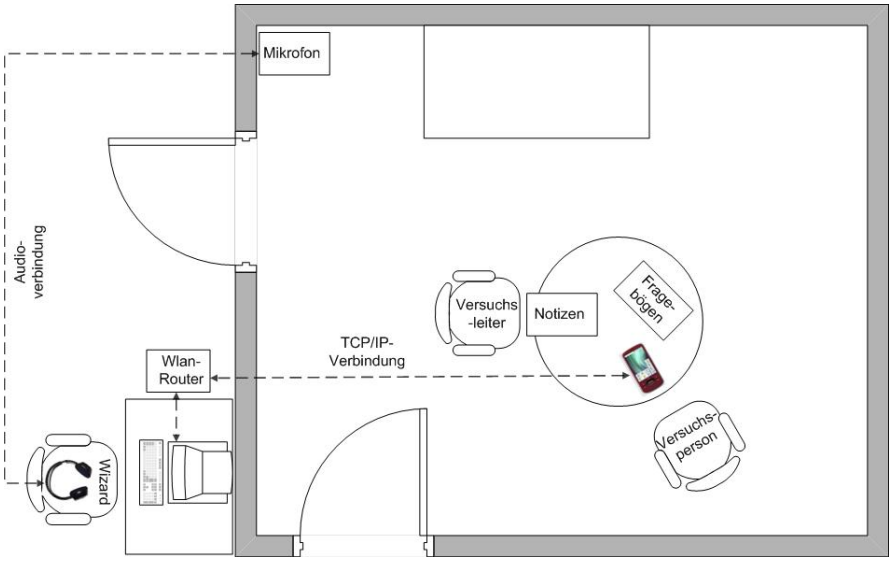

Fig. A.8 Setup of the experiments 1 and 2.

A.9 Setup of Experiment 3

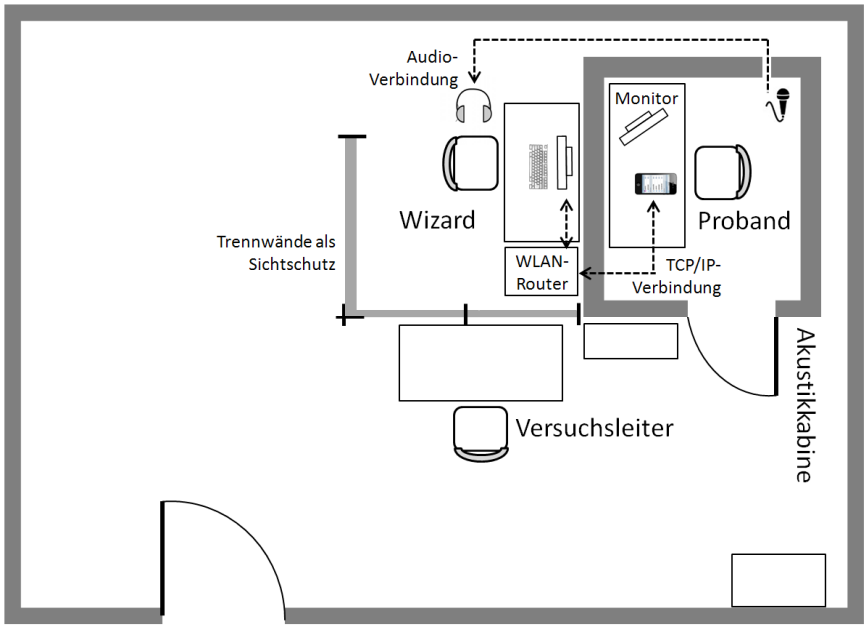

Fig. A.9 Setup of experiment 3.

A.10 Statement of Agreement

Einverständniserklärung

Hiermit erkläre ich mich einverstanden, an der Studie

Restaurantbuchungssystem

teilzunehmen. Für meine Teilnahme erhalte ich eine Vergütung von 10 €.

Die in diesem Zusammenhang erhobenen Daten werden aufgezeichnet, in anonymisierter Form gespeichert und zu wissenschaftlichen Zwecken ausgewertet.

Meine Daten werden streng vertraulich behandelt und können nur von entsprechend autorisierten Personen eingesehen werden.

Meine Teilnahme an der Studie erfolgt freiwillig.

Mir ist bewusst, dass ich die Studie zu jedem Zeitpunkt abbrechen kann. In diesem Fall verliere ich jedoch den Anspruch auf die oben genannte Vergütung.

Ort, Datum _____ Unterschrift _____

Fig. A.10 Participants of all experiments had to sign the statement of agreement.

A.11 Instructions experiment 1 and 2

Lieber Teilnehmer,

Vielen Dank, dass Sie sich die Zeit für diesen Test nehmen. Im Folgenden werden Sie gebeten den Prototypen eines Restaurantbuchungssystems für Mobiltelefone zu testen. Mit Hilfe des Systems ist es möglich ein Restaurant einer bestimmten Kategorie in einer Zielstadt zu suchen. Dazu müssen folgende Eingaben getätigt werden:

Die Stadt in der das Restaurant liegen soll
Die Kategorie des Restaurants (z.B. asiatisch oder italienisch)
Die Tageszeit zu der ein Tisch benötigt wird
Die Anzahl der Personen für die ein Tisch benötigt wird

Für den Test bitten wir Sie insgesamt 15 Restaurants zu suchen. Nach der Suche eines jeden Restaurants bitten wir Sie einen kurzen Fragebogen auszufüllen.

Alle Eingaben können entweder über einen Touchscreen oder über direkte Spracheingabe getätigt werden. Für die Sprachsteuerung stehen Ihnen Kommandos in deutscher Sprache zur Verfügung. Sie sollten dem System klare Befehle geben, die nur aus wenigen oder einzelnen Wörtern bestehen.
Bitte Verwenden Sie bei getippten Eingaben nur die Funktionen des Touchscreens und nicht die Knöpfe am unteren Rand des Telefons.

Uns sind Ihr spontanes Verhalten und Ihre Meinung über das System wichtig. Wir wollen nicht Sie testen, sondern unser System. Wenn es Schwierigkeiten mit der Bearbeitung der Aufgaben gibt, ist das kein Fehler Ihrerseits, sondern ein Problem unseres Systems.
Sie helfen uns mit diesem Test bei der Weiterentwicklung, daher sind wir dankbar für jeden Fehler, den Sie finden. Alle während des Versuchs aufgenommenen Daten werden vertraulich behandelt und ohne Bezug zu Ihrer Person ausgewertet.
Bei Fragen oder Unklarheiten können Sie sich jederzeit an die Versuchsleiter wenden.

Im Folgenden wird der Versuchsleiter mit Ihnen einen Fragebogen ausfüllen Sie mit dem Gerät und den Eingabemöglichkeiten vertraut machen.

Fig. A.11 Instructions for the participants of experiment 3. The experiment was conducted in two blocks. In the scope of this work only data from the first block is used.

A.12 Instructions experiment 3

Herzlich Willkommen! Vielen Dank, dass Sie sich die Zeit für diese Untersuchung nehmen.

Sie werden heute den mobilen Prototypen eines Restaurantbuchungssystems testen. Mit diesem Prototypen können Sie online einen Tisch in einem bestimmten Restaurant reservieren.

Die Suche nach einem Restaurant erfolgt über folgende **vier Kriterien:**

- die Stadt, in der ein Restaurant liegen soll (z.B. Berlin)
- die Kategorie des Restaurants (z.B. italienisch)
- die Uhrzeit, zu der ein Tisch gewünscht wird (z.B. 18:00 Uhr) und
- die Anzahl der Personen, für die ein Tisch benötigt wird (z.B. zwei Personen).

Für den Test geben wir Ihnen zwei Blöcke mit je **12 Bedienaufgaben** vor, die Ihnen einzeln auf dem Bildschirm präsentiert werden. Jede Aufgabe nennt Ihnen die jeweiligen Eingaben, die Sie für eine Restaurantsuche vornehmen sollen, z.B. „ein Restaurant mit italienischer Küche in Berlin um 18:00 Uhr für zwei Personen".

Nach jeder Aufgabe werden Sie die Interaktion auf einem kurzen Fragebogen bewerten.

Die Bedienung des Prototypen erfolgt wahlweise über Touchscreen oder über Sprache. Sie selbst entscheiden, wie sie das System bedienen möchten, und Sie können *jederzeit* – auch innerhalb einer Restaurantsuche – zwischen den Eingabeformen wechseln.

Zur Eingabe über **Touchscreen** klicken Sie bitte mit dem Finger auf die entsprechenden Buttons im Display des Prototypen (siehe Startbildschirm in Abbildung 1).

Die Bedienung über **Spracherkennung** erfolgt mittels Sprachkommandos, die im Wesentlichen den Displaybeschriftungen entsprechen, z.B. „Stadt auswählen", „Berlin" oder „Restaurant suchen".
Bitte beachten Sie bei der Spracherkennung, dass Sie auf dem Startbildschirm (Abbildung 1) zunächst das jeweilige Kriterium nennen müssen, das Sie auswählen möchten (also z.B. „Stadt auswählen"), bevor Sie die eigentliche Eingabe, z.B. „Mannheim" direkt nennen können. Das gleiche gilt für die anderen drei Kriterien Kategorie, Uhrzeit und Personenanzahl.

Abbildung 1: Startbildschirm des Prototypen

Haben Sie alle vier Kriterien erfolgreich eingegeben, schicken Sie bitte eine Reservierungsanfrage über „Restaurant suchen" ab. Damit ist eine Testaufgabe erledigt.

Sie werden die Bedienung des Prototypen nun zunächst an einigen Beispielaufgaben kennenlernen. **Haben Sie vorab noch Fragen zum Versuch oder zur Bedienung?**

Fig. A.12 Instructions for the participants of experiment 3. The experiment was conducted in two blocks. In the scope of this work only data from the first block is used.

A.13 Social demographic Questionnaire

Geschlecht: O weiblich O männlich

Alter: _____

Beruf (wenn Student, mit Fachrichtung): _____

Haben Sie in der Vergangenheit bei Versuchen mit Sprach-steuerungssystemen teilgenommen?	O ja O nein
Besitzen Sie ein Handy mit Touchscreen Funktionalität?	O ja O nein
Wie oft nutzen Sie Touchscreen Eingabesysteme (z.B. Smartphones, Fahrkartenautomaten, Bankautomaten, u.ä.)?	O täglich O wöchentlich O seltener O nie
Hat ihr Handy eine Sprachfunktion?	O ja O nein
Wenn ja, benutzen Sie diese?	O immer O oft O selten O nie
Sprechen Sie auf Anrufbeantworter/Mailboxes?	O immer O oft O selten O nie
Haben Sie Erfahrungen mit Sprachdialogsyste-men/Spracheingabesystemen? (z.B. automatische Hotlines der Bahn, von Versicherungen, Telefonanbietern, Navi-Eingabe über Sprache u.ä.)	O ja O nein
Wie oft nutzen Sie Sprachdialogsysteme?	O täglich O wöchentlich O seltener O nie

Fig. A.13 Participants of all experiments had to fill out a social demographic questionnaire.

A.14 Tasks and task construction

A.14.1 Training Tasks of Experiment 2

1. Bitte suchen Sie ein Fischrestaurant in Kiel, ab 20:00 Uhr fr 10 Personen.
2. Bitte suchen Sie ein Sushi-Restaurant in Mnchen, ab 21:00 Uhr fr zwei Personen.
3. Bitte suchen Sie ein orientalisches Restaurant in Dortmund, ab 13:00 Uhr fr 18 Personen.

The first task processing was carried out unimodal only via the touch screen. The second task processing was carried out unimodal only via speech input. The third task processing was carried out multimodal. The participants could at any time choose between touch screen and speech input.

A.14.2 Target Trials of Experiment 1 and 2

The order of the target trials was systematically varied. Thus, each participant had an individual task order.

1. Bitte suchen Sie ein Restaurant mit brasilianischer Kche in Augsburg, ab 12:00 Uhr fr zwei Personen.
2. Bitte suchen Sie ein Bio-Restaurant in Berlin, ab 16:00 Uhr fr vier Personen.
3. Bitte suchen Sie ein Restaurant mit chinesischer Kche in Dortmund, ab 18:00 Uhr fr fnf Personen.
4. Bitte suchen Sie ein Restaurant mit amerikanischer Kche in Dsseldorf, ab 17:00 Uhr fr drei Personen.
5. Bitte suchen Sie ein Restaurant mit deutscher Kche in Bremen, ab 13:00 Uhr fr vier Personen.
6. Bitte suchen Sie ein Restaurant mit griechischer Kche in Erfurt, ab 20:00 Uhr fr neun Personen.
7. Bitte suchen Sie ein Restaurant mit italienischer Kche in Kln, ab 0:00 Uhr fr vierzehn Personen.
8. Bitte suchen Sie ein Fischrestaurant in Hamburg, ab 21:00 Uhr fr zehn Personen.
9. Bitte suchen Sie ein Restaurant mit indischer Kche in Frankfurt, ab 20:00 Uhr fr zehn Personen.
10. Bitte suchen Sie ein Restaurant mit japanischer Kche in Hannover, ab 22:00 Uhr fr 13 Personen.
11. Bitte suchen Sie ein Restaurant mit mexikanischer Kche in Leipzig, ab 07:00 Uhr fr 17 Personen.
12. Bitte suchen Sie ein Sushirestaurant in Stuttgart, ab 11:00 Uhr fr 22 Personen.
13. Bitte suchen Sie ein Restaurant mit portugiesischer Kche in Mnchen, ab 10:00 Uhr fr 18 Personen.

14. Bitte suchen Sie ein Restaurant mit russischer Kche in Mannheim, ab 11:00 Uhr fr 20 Personen.
15. Bitte suchen Sie ein Restaurant mit mediterraner Kche in Rostock, ab 11:00 Uhr fr 19 Personen.

In experiment 1 the participants did not perform explicit training trials. Therefore, the first three tasks of target trials were treated tasks as training trials, and therefore left out in the calculation of the average modality selection.

A.14.3 Training trials of experiment 3

1. Suchen Sie ein Fischrestaurant in Kiel ab 20:00 Uhr für 10 Personen.
2. Suchen Sie ein Sushi-Restaurant in Wiesbaden ab 21:00 Uhr für 2 Personen.
3. Suchen Sie ein orientalisches Restaurant in Dortmund ab 13:00 Uhr für 18 Personen.

The first task processing was carried out unimodal only via the touch screen. The second task processing was carried out unimodal only via speech input. The third task processing was carried out multimodal. The participants could at any time choose between touch screen and speech input.

A.14.4 Target trials of experiment 3

The order of the target trials was systematically varied. Thus, each participant had an individual task order.

1. Suchen Sie ein Restaurant mit amerikanischer Küche in Freiburg ab 16:00 Uhr für 13 Personen.
2. Suchen Sie ein Restaurant mit griechischer Küche in Erfurt ab 20:00 Uhr für 9 Personen.
3. Suchen Sie ein Restaurant mit mediterraner Küche in Bremen ab 10:00 Uhr für 6 Personen.
4. Suchen Sie ein Restaurant mit chinesischer Küche in Dortmund ab 18:00 Uhr für 5 Personen.
5. Suchen Sie ein Restaurant mit indischer Küche in München ab 01:00 Uhr für 21 Personen.
6. Suchen Sie ein Restaurant mit portugiesischer Küche in Stuttgart ab 11:00 Uhr für 22 Personen.
7. Suchen Sie ein Restaurant japanischer Küche in Düsseldorf ab 22:00 Uhr für 4 Personen.
8. Suchen Sie ein Restaurant mit brasilianischer Küche in Augsburg ab 12:00 Uhr für 2 Personen.

9. Suchen Sie ein Restaurant mit deutscher Küche in Rostock ab 13:00 Uhr für 18 Personen.
10. Suchen Sie ein Restaurant mit italienischer Küche in Kln ab 00:00 Uhr für 14 Personen.
11. Suchen Sie ein Restaurant mit russischer Küche in Hannover ab 06:00 Uhr für 10 Personen.
12. Suchen Sie ein Restaurant mit mexikanischer Küche in Leipzig ab 07:00 Uhr für 17 Personen.

A.15 Examination of the statistical criteria

A.15.1 Experiment 1

A.15.1.1 Distribution form of the dependent variable

Table A.1 Characteristic values for checking the distribution of the dependent variables. The table gives the descriptive characteristics of the distributions and sizes for testing the normal distribution using the Shapiro-Wilk test; $^{*}p < .05$.

Dependent variable	*M*	*s*	*Skew*	*Kurtosis*	*Shapiro-Wilk*	*df*	*p*
Speech LD 1	.56	0.37	-0.19	-1.45	.879	16	$< .000^{*}$
Speech LD 2	.78	0.32	-1.30	0.29	.721	16	$< .000^{*}$
Speech LD 3	.88	0.25	-2.57	6.26	.563	16	$< .000^{*}$
Speech LD 4	.89	0.25	-2.45	6.01	.603	16	$< .000^{*}$
Speech LD 5	.90	0.23	-2.71	9.28	.641	16	$< .000^{*}$
Speech LD 6	.92	0.22	-2.84	9.83	.507	16	$< .000^{*}$

A.15.1.2 Homogeneity of error variances (Levene test)

Table A.2 Levene test to check the homogeneity of variances; $^{*}p < .05$.

Dependent variable	*F*	*df1*	*df2*	*p*
Speech LD 1	5.709	3	14	$.002^{*}$
Speech LD 2	6.704	3	14	$.001^{*}$
Speech LD 3	6.782	3	14	$.001^{*}$
Speech LD 4	2.126	3	14	.111
Speech LD 5	1.660	3	14	.189
Speech LD 6	1.304	3	14	.285

A.15.2 Experiment 2

		Speech LD1	Speech LD2	Speech LD3	Speech LD4	Speech LD5	Speech LD6
N		29	29	29	29	29	29
Parameter der Normalverteilung[a,b]	Mittelwert	,1767	,5288	,7243	,7705	,7935	,9037
	Standardabweichung	,25372	,30144	,27479	,23875	,26988	,18209
Extremste Differenzen	Absolut	,312	,142	,187	,177	,226	,322
	Positiv	,312	,133	,158	,168	,222	,298
	Negativ	-,243	-,142	-,187	-,177	-,226	-,322
Kolmogorov-Smirnov-Z		1,678	,764	1,007	,951	1,218	1,735
Asymptotische Signifikanz (2-seitig)		,007	,604	,263	,326	,103	,005

a. Die zu testende Verteilung ist eine Normalverteilung.

b. Aus den Daten berechnet.

Fig. A.14 Kolmogorov-Smirnov Test. Distribution form of the dependent variable.

A.15.3 Experiment 3

A.15.3.1 Distribution form of the dependent variable

Table A.3 Characteristic values for checking the distribution of the dependent variables. The table gives the descriptive characteristics of the distributions and sizes for testing the normal distribution using the Shapiro-Wilk test; $^{*}p < .05$.

Dependent variable	*M*	*s*	*Skew*	*Kurtosis*	*Shapiro-Wilk*	*df*	*p*
Speech LD 1	.58	0.36	-0.18	-1.49	.878	48	$< .000^{*}$
Speech LD 2	.76	0.30	-1.30	0.61	.776	48	$< .000^{*}$
Speech LD 3	.81	0.27	-1.68	2.26	.764	48	$< .000^{*}$
Speech LD 4	.85	0.25	-2.17	4.37	.645	48	$< .000^{*}$
Speech LD 5	.84	0.27	-1.79	2.80	.728	48	$< .000^{*}$
Speech LD 6	.86	0.27	-2.02	3.52	.664	48	$< .000^{*}$

A.15.3.2 Homogeneity of error variances (Levene test)

Table A.4 Levene test to check the homogeneity of variances; $^{*}p < .05$.

Dependent variable	*F*	*df1*	*df2*	*p*
Speech LD 1	1.462	3	44	.238
Speech LD 2	7.481	3	44	$.000^{*}$
Speech LD 3	5.879	3	44	$.002^{*}$
Speech LD 4	2.466	3	44	.075
Speech LD 5	2.798	3	44	.051
Speech LD 6	4.792	3	44	$.006^{*}$

Appendix B
MeMo Modelling Details

B.1 RBS MeMo System Model

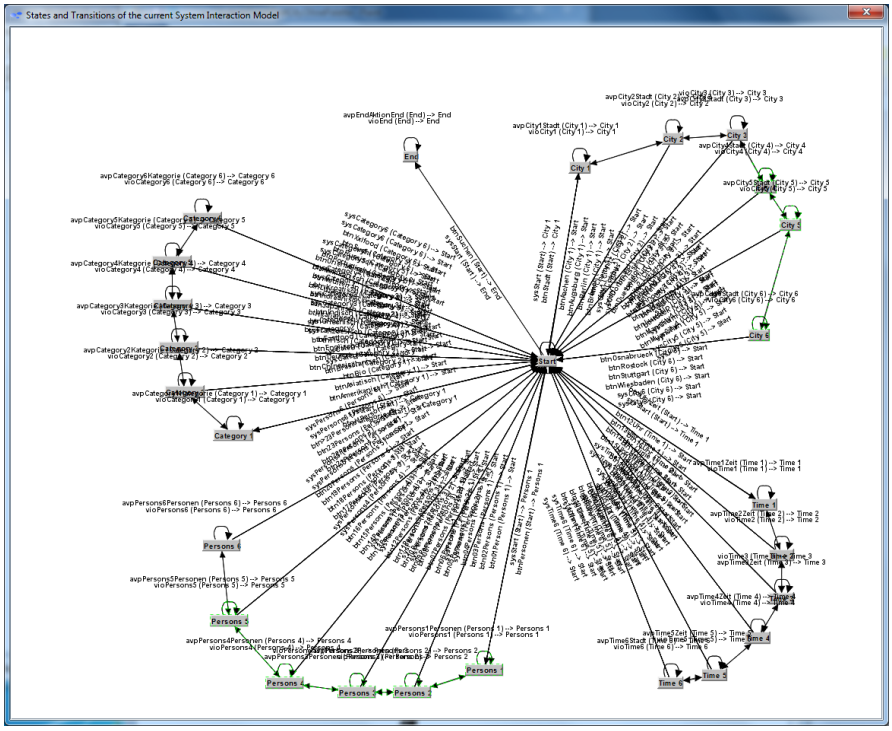

Fig. B.1 The RBA system model.

B.2 RBS MeMo System Model Detail

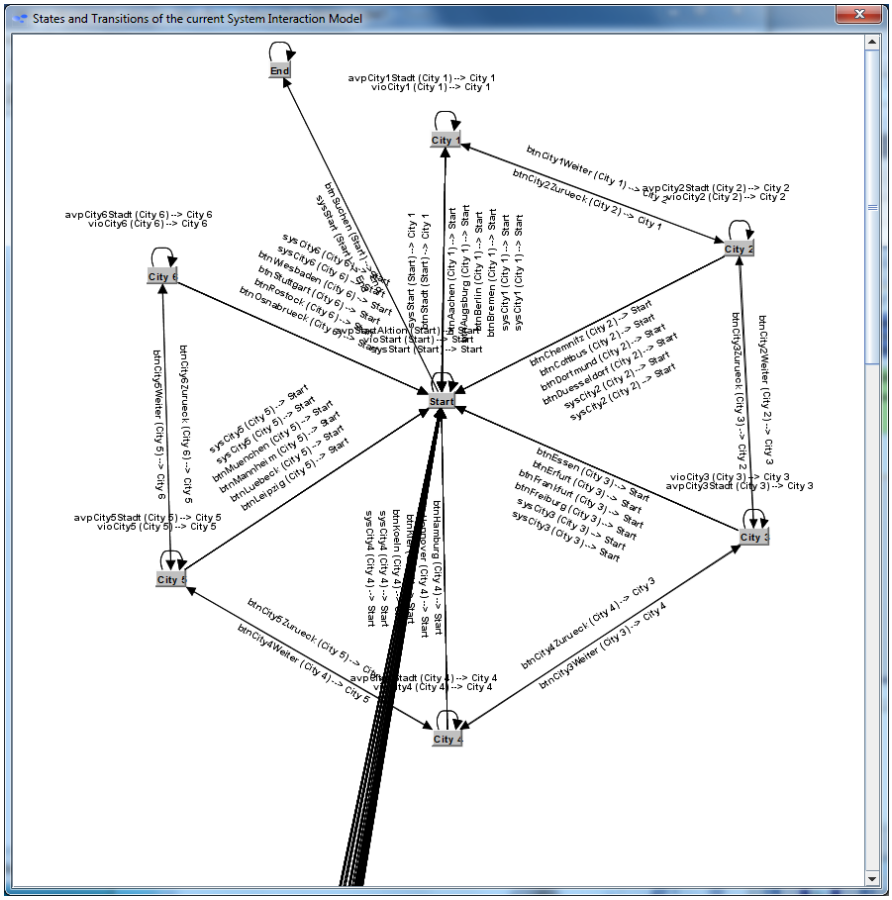

Fig. B.2 Detail ”city list” of the RBA system model.

B.3 MeMo Default User Model

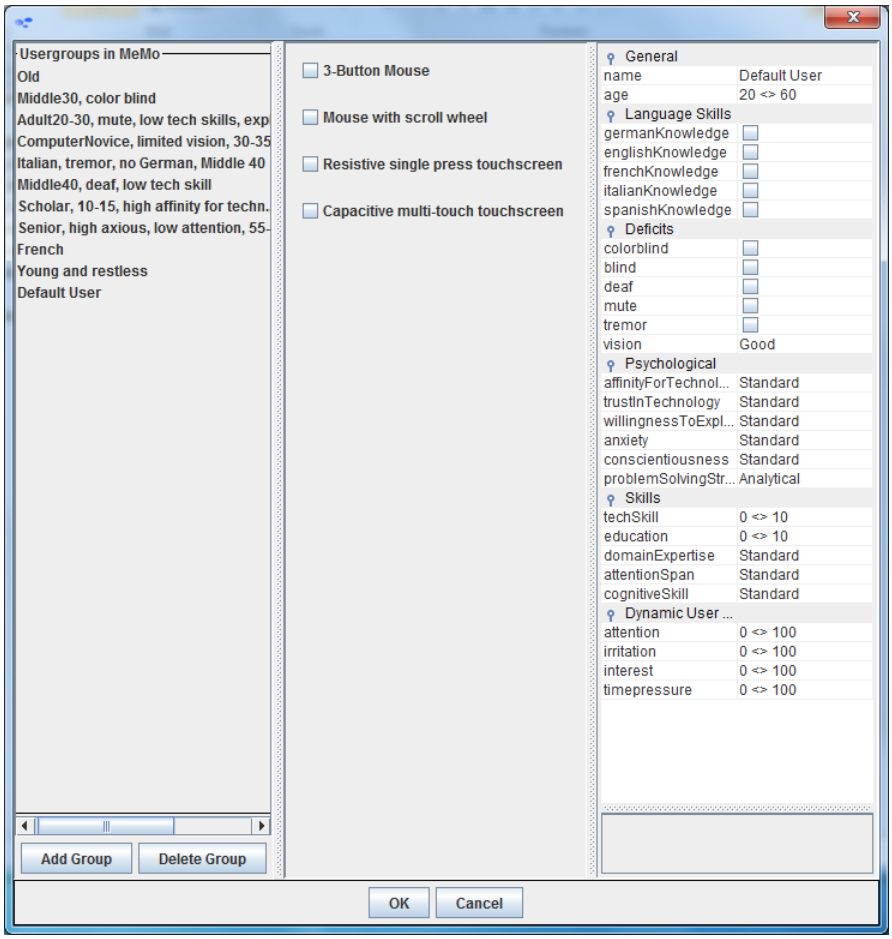

Fig. B.3 The default user model in the user model designer.

B.4 MeMo HCI Swoosher Properties

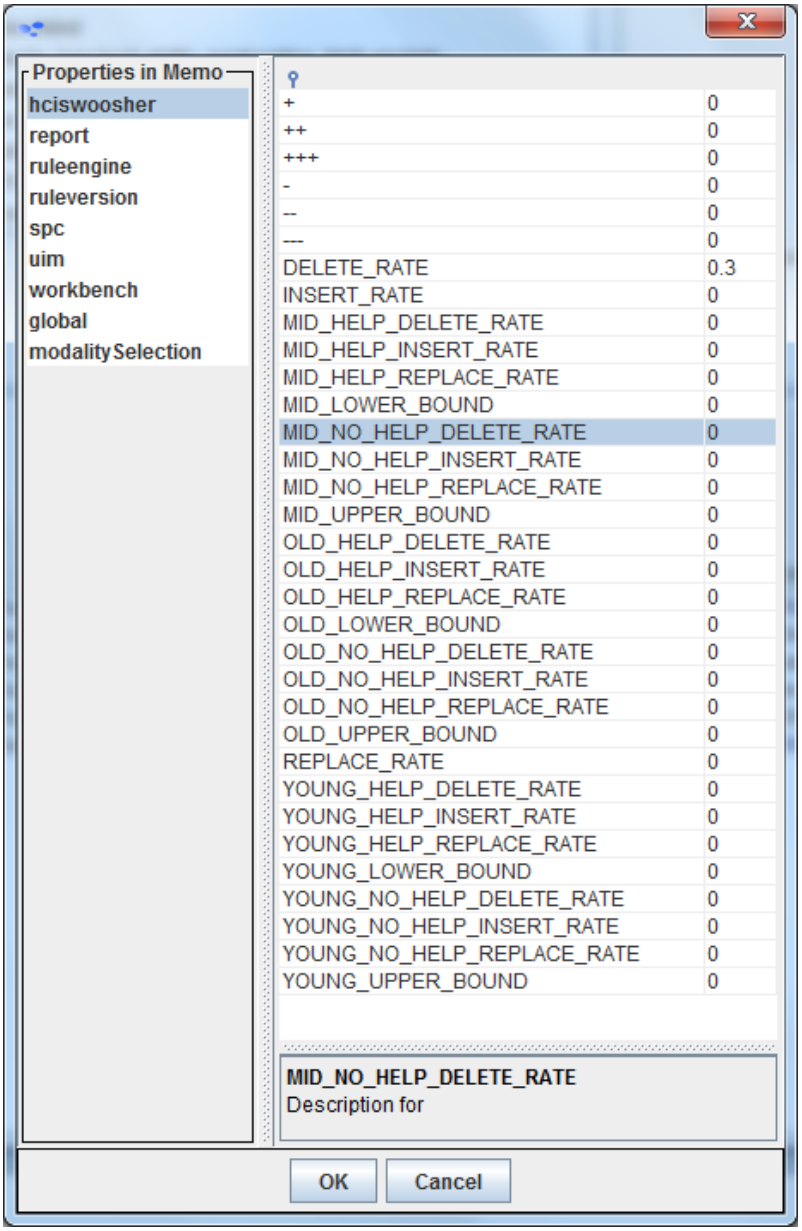

Fig. B.4 HCI swoosher properties.

B.5 MeMo Modality Selection Properties

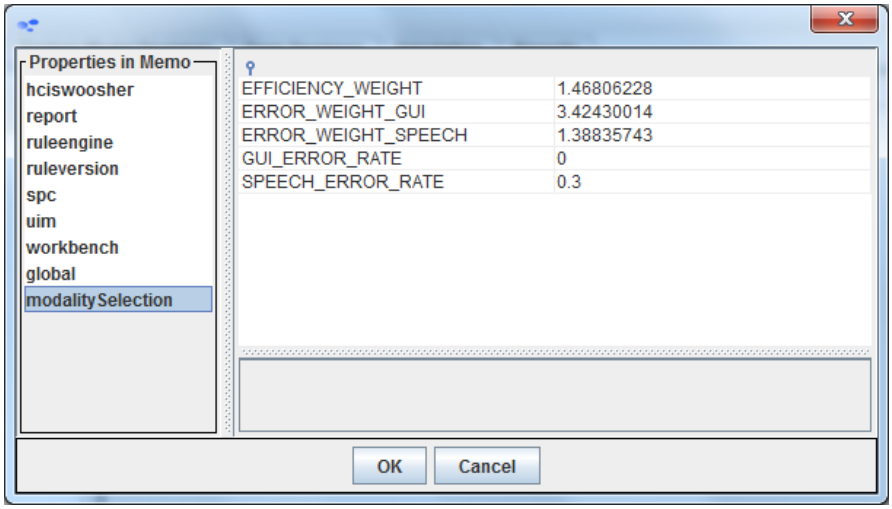

Fig. B.5 Modality selection properties.

B.6 MeMo Solution Path Calculator Properties

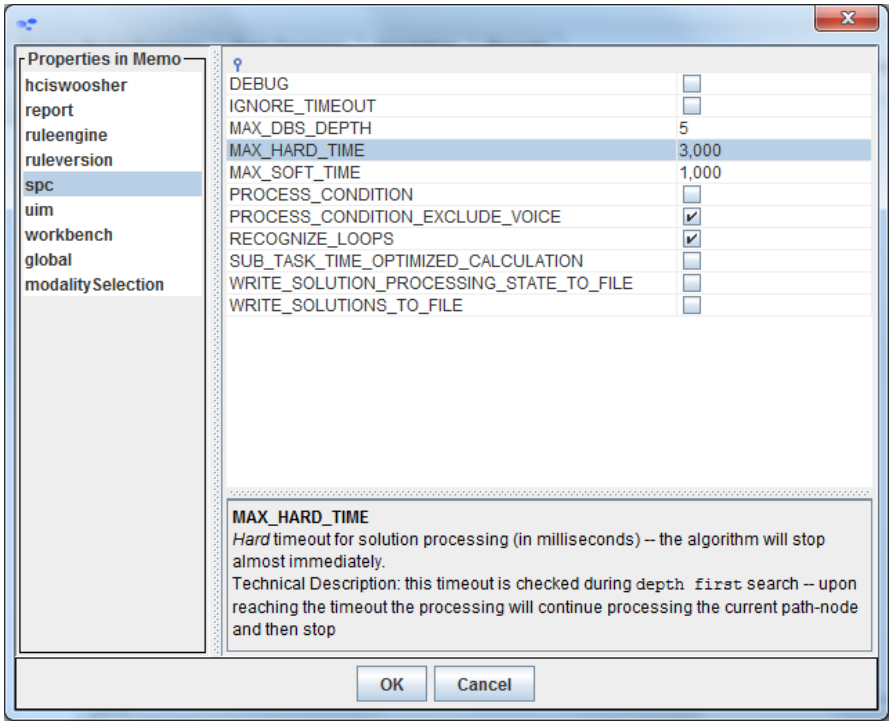

Fig. B.6 Solution path calculator properties.

B.7 MeMo User Interaction Model Properties

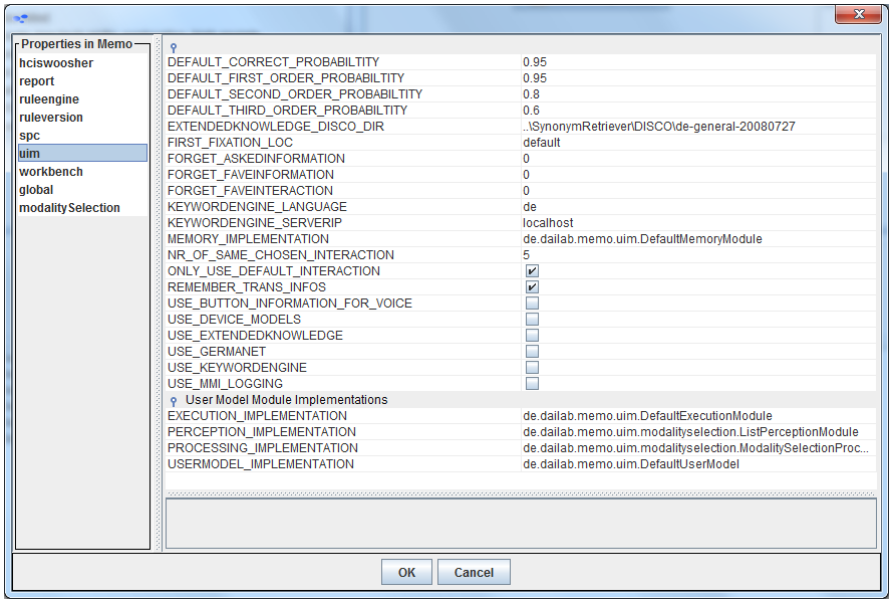

Fig. B.7 User interaction model properties.

B.8 MeMo Reports - with low Interaction Probability

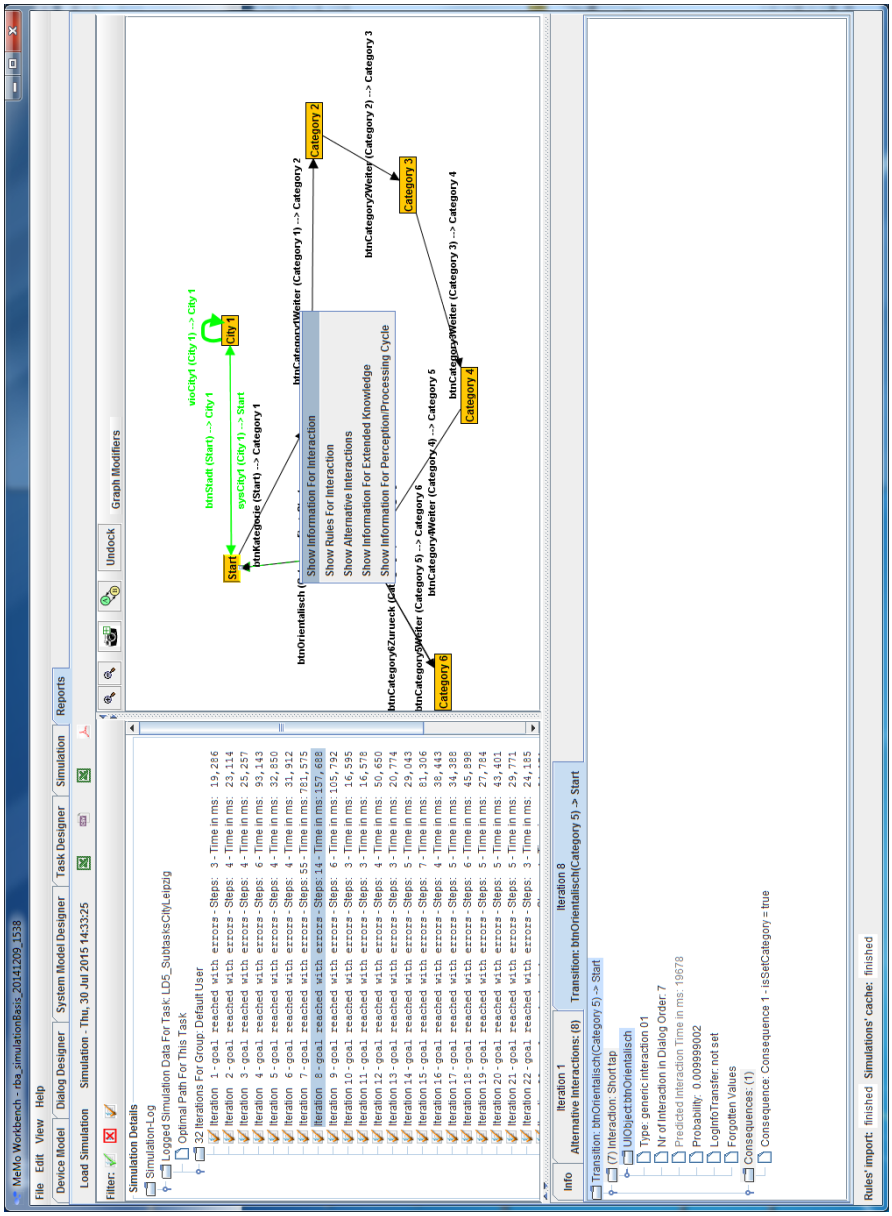

Fig. B.8 A MeMo report showing a low probability for an interaction.

B.9 MeMo Reports - with high Interaction Probability

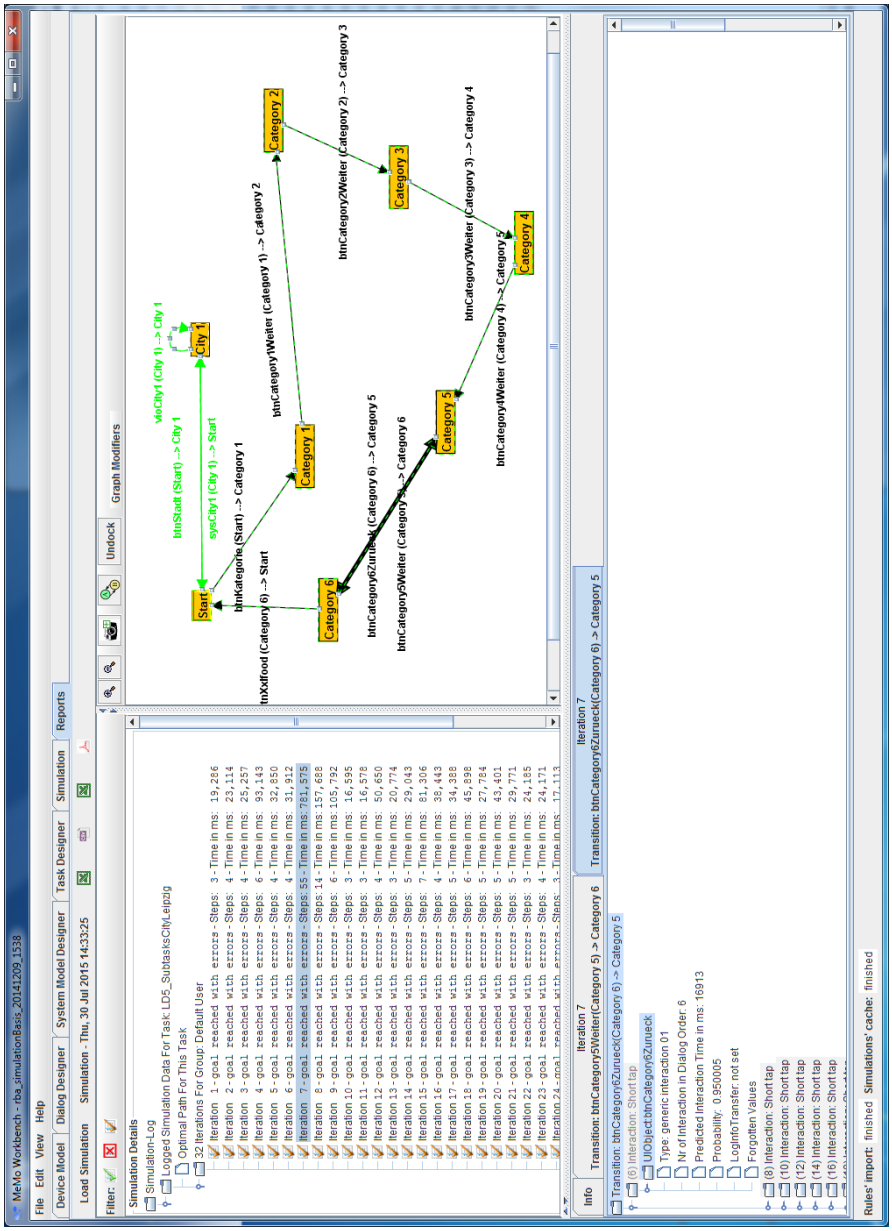

Fig. B.9 A MeMo report showing a high probability for an interaction.

Appendix C
CogTool Modelling Details

C.1 Lisp Implementation of the Modality Selection Algorithm

Listing C.1 Think modality selection ACT-R production generated from the CogTool script

```
1  ;;; The ACT-R model in this file is merged from two models. The
       original model
2  ;; performed GUI interaction only. The other model also performed
       VUI input.
3  ;; After merging the models state slots inside the productions
       were adapted, so
4  ;; that both modalities are available for system input.
5  ;; The function below is used to select one of the modalities.
6  ;; Stefan Schaffer
7  (setf *random-state* (make-random-state t))
8  (defun select-mod (i-v i-t e-v e-t)
9          (let (g-v g-t c a b p)
10                 (setf g-v 1.24106238)
11         (setf g-t 3.18078481)
12         (setf c 1.49336935)
13                 (setf a (* (* i-t i-t) (+ 1 (* -1 e-v) (* -1 (*
                       e-v g-v)) (* (* e-v e-v) g-v))))
14                 (setf b (* (* i-v i-v) (+ 1 (* -1 e-t) (* -1 (*
                       e-t g-t)) (* (* e-t e-t) g-t))))
15         (setf p (/ 1 (+ 1 (* c (/ b a)))))
16                 (if (>(random 1.0) p) "touch" "voice")))
17
18 (defun do-it (n)
19         (dotimes (i n)
20                 (cogtool-run-model)
21                 (reset)))
```

C.2 The CogTool Design of the RBA

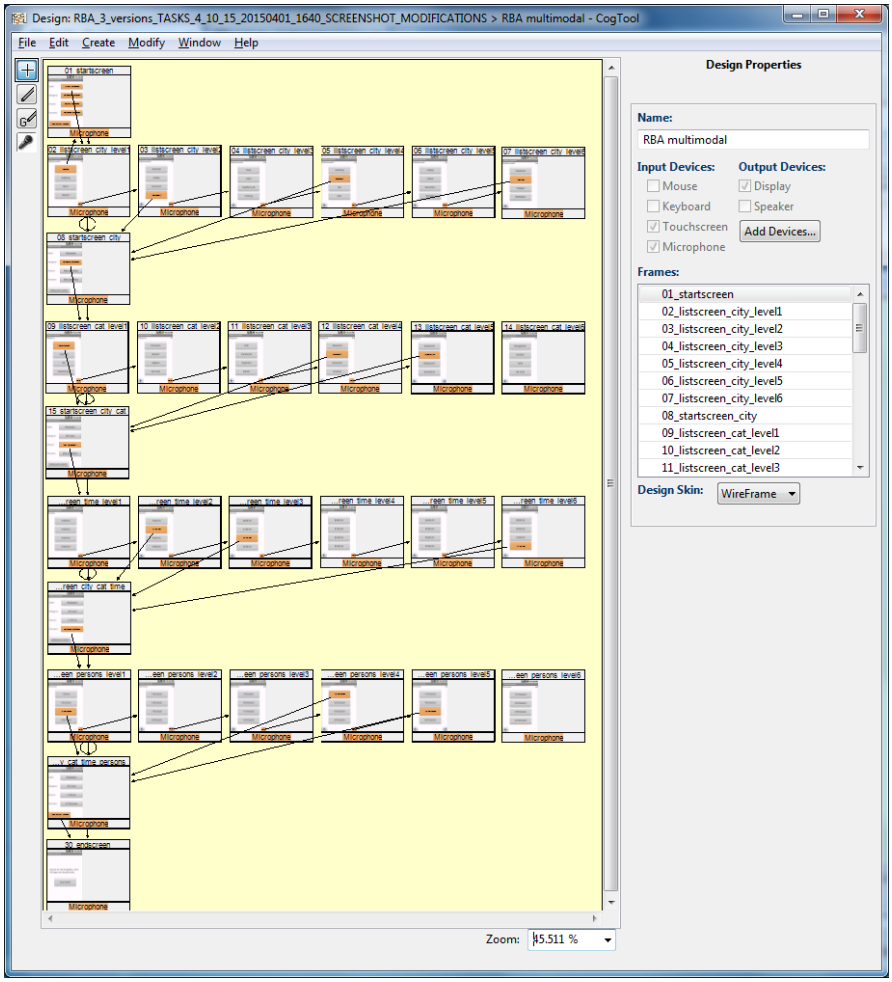

Fig. C.1 Graph of the multimodal CogTool RBA design.

C.3 CogTool Project Window

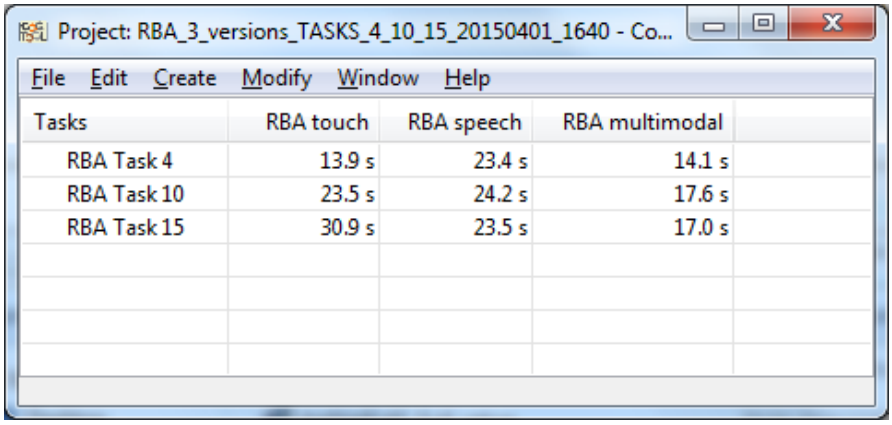

Fig. C.2 CogTool project window - overview over designed systems, demonstrated tasks, and performance predictions.

C.4 Results of the CogTool Ppredictions

Table C.1 Results of the CogTool predictions.

Condition	Tasks	RBA		
		touch	speech	multimodal
CT_{ms}	4	13.9	23.4	14.1
CT_{ms}	10	23.5	24.2	17.6
CT_{ms}	15	30.9	23.5	17.0
$CT_{default}$	4	18.5	16.2	14.2
$CT_{default}$	10	32.7	17.0	16.4
$CT_{default}$	15	43.5	16.3	15.8