Situation- and User-Adaptive Dialogue Management



ulm university universität ulm

Gregor Bertrand

geboren in Ravensburg

Institute of Communications Engineering: Dialogue Systems

University of Ulm

Dissertation zur Erlangung des Doktorgrades Dr.rer.nat. der Fakultät für Ingenieurwissenschaften und Informatik der Universität Ulm

2014

Officiating Dean: Prof. Dr. Tina Seufert

1. Reviewer: Prof. Dr. Dr.-Ing. Wolfgang Minker

2. Reviewer: Prof. Dr. Harald C. Traue

Day of the defense: 24th of February, 2015

Abstract

In the sea of human machine interaction a growing trend to personalize the interaction experience has been surfacing over the last decades. Human machine interfaces have more and more become companion systems which on the one hand collect data about the user either explicitly or implicitly and on the other hand adapt their interface and their behaviour to the context of the user.

This trend has also found its way into the world of dialogue systems.This thesis addresses different aspects this trend brings about for the central part of any dialogue system, the dialogue manager. However, it focuses on the reaction of the dialogue manager to changing context information and investigates the technical side, as well as the architectural and interface side. As the detection of user context information has been and still is extensively investigated, it leaves out the part of gathering contextual information and assumes them as given.

Firstly, context information which may be available is discussed presenting the aspects of gathering, processing and reacting to it. Secondly, the special topic of user emotion as context information is explored, especially focusing on the aspect of reacting to it. Then, an approach for integrating adaptivity into dialogue management in the areas of model, control, and design is presented. Finally, this approach is evaluated in a proof-of-concept manner for multimodal adaptivity and adaptivity of the dialogue itself as well as in an user study for several examples of dialogue adaptivity.

Deutschsprachige Zusammenfassung

Bei dem Versuch, dem Nutzer den Umgang mit technischen Geräten zu erleichtern, ist man dazu übergegangen, immer mehr die persönlichen Vorlieben und Gewohnheiten des Nutzers in das Design der Schnittstelle einfliessen zu lassen. Das bedeutet, dass nicht nur allgemeine für alle Nutzer gültige Designrichtlinien beachtet werden, sondern dass auch auf den Nutzer individuell eine angepasste Schnittstelle erstellt wird. Im Zuge dieser nutzerzentrierten Schnittstellenentwicklung wurden nicht nur langfristigere Veränderungen der Schnittstelle wie das Arrangement der Menüführung nach dem Geschmack des Nutzers vorgesehen. Es wird immer mehr darüber diskutiert, wie kurzfristige Veränderungen die Interaktion mit dem Nutzer verbessern können.

Insbesondere die Berücksichtigung der Emotion des Nutzers im Umgang mit dem technischen Gerät rückte bei dieser Betrachtung stärker in den Vordergrund. Die vorliegende Dissertation befasst sich mit dem Thema, inwiefern Dialogsysteme adaptiv sein können mit dem Fokus auf der technischen Realisation der Adaptivität und der Untersuchung des Nutzens von adaptiven Dialogsystemen. Ein Aspekt ist auch in diesem Zusammenhang, wie Adaptivität auf Nutzeremotion in Dialogsystemen realisiert werden kann.

Hierzu wurden in zwei Vorstudien Emotion in der Interaktion des Nutzers mit Dialogsystemen prospektiv in Zusammenarbeit mit der Sektion Medizinische Psychologie des Universitätsklinikums Ulm untersucht.

Auf den gewonnenen Erkenntnissen wurde ein Dialogsystem konzipiert. Zum Umfang des prototypisch umgesetzten Dialogsystems gehörten sowohl ein Ansatz zur effizienten Modellierung von Adaptivität, ein hierarchisches Dialogmodell zur Abbildung des adaptiven Dialogs als auch ein neuartiger Algorithmus zur Traversierung des Modells und zur Steuerung des Dialogs. Im Anschluss an die Umsetzung wurde das System in mehreren Szenarien evaluiert.

Im Rahmen des Ansatzes zur effizienten Modellierung von Adaptivität wurde ein graphischer Editor entwickelt, der die Erstellung von Dialogen vereinfacht und das Design des Dialogs vom eigentlichen Algorithmus entkoppelt. Dadurch wird ohne tiefes Wissen über die Details der Verarbeitung und beim ungleich komplexen Erstellen von adaptiven Dialogen Arbeitsteilung möglich. So kann ein Designer ohne tiefes Wissen über die Algorithmik der Verarbeitung mehr auf das Schnittstellendesign fokussiert arbeiten.

Das hierarchische Dialogmodell ermöglicht die Modellierung von benutzer- / emotionsangepassten Varianten des Dialogs, die dann zur Laufzeit abhängig vom Nutzerzustand angesteuert werden können. Dem angestiegenen Umfang (durch die verschiedenen Varianten) des Modells wird Rechnung getragen, indem das Dialogmodell sich über mehrere Dateien dank der Hierarchie gleich einem Ordnerdateisystem aufteilen lässt. Des weiteren beschränkt sich das Modell nicht auf die unimodale Dialoge, sondern ist so konzipiert, dass Multimodalität problemlos möglich ist.

Die Steuerung / Algorithmik zur Verarbeitung des Dialogs lehnt sich an Prinzipien an, die aus der Constraint Programmierung stammen und verwendet werden, um in Situationen, die durch viele Parameter beeinflusst werden, deterministisch eine geeignete Entscheidung (bzw. Belegung der gegebenen Variablen) zu finden. Bisher wurde ein solcher Ansatz noch nicht im Zusammenhang mit Dialogsystemen etabliert. Es gibt lediglich mehrere Systeme, die logische Programmiersprachen (wie z. B. Prolog), welche als Vorstufe für Constraint Programmiersprachen gesehen werden können, verwenden. Der Nachteil an diesen Systemen ist neben der gerin-

geren Leistungsfähigkeit, die umständlichere Modellierung, die beim vorliegenden Prototypen bereits durch einen graphischen Editor gelöst ist.

Die Evaluation des Systems beschränkte sich nicht nur auf die Vorstudien, sondern umfasste auch die Einbindung in einen multimodalen Kontext im Rahmen des SFB-TRR 62 und eine Umsetzung einer Sprachdialoganwendung. Die Einbindung in den multimodalen Kontext wurde in der erfolgreichen Begehung der Entwicklung durch ein wissenschaftliches Kommitee des Landes evaluiert, und trug zur gewährten Weiterfinanzierung der Forschung des SFB-TRR 62 bei. Die Sprachdialoganwendung wurde in einer Nutzerstudie in Zusammenarbeit mit der Lehr- und Lernforschung des Instituts für Psychologie und Pädagogik evaluiert.

In dem genannten Forschungsfeld sind einige Fragen noch immer nicht geklärt. So hat zum Beispiel die Psychologie noch keine letztendliche Antwort auf die Frage gefunden, was Emotionen denn sind und auch im Umgang mit diesen gibt es widersprüchliche Ansätze. Es besteht hier noch Grundsatzforschungsbedarf und das bedeutet wiederum dass auch die vorliegende Arbeit in diesem Punkt nur vorläufige Antworten anbieten kann. Im Punkt der technischen Realisierung ist die Arbeit deshalb offen gehalten und bietet Anknüpfungspunkte für Erweiterungen und verschiedenartige Umsetzungen eines konkreten Dialogsystems.

Acknowledgements

Let's start with the professional surroundings which fostered my research work. To begin with I would like to thank my professor Wolfgang Minker who supported me throughout the thesis. Many opportunities for interesting conferences and learning became possible because of him. His mentoring was always friendly, fair, and provided the neccessary time for advancement with the research.

Then I would like to mention the support of professor Harald Traue. The cooperation with his research group generated many interesting new influences and aspects for the thesis. In this group I would like to mention Holger Hoffmann, David Hrabal, Stefanie Rukavina, and Steffen Walter personally, but also the many people who supported the research and who I may have forgotten here.

Furthermore, the SFB-TRR62 surroundings provided many opportunities for learning (lectures, seminars, meetings, etc.) which enriched my perspective on the field of companion technologies in general and especially adaptivity in human machine interaction. I would like to thank all professors who made this research possible through their contribution.

In addition, I would like to mention the members of my working group who shared their ideas and suggestions with me and in this way were a big improvement to the thesis. Thank you, Amparo Albalate, Tatiana Gasanova, Tobias Heinroth, Hansjörg Hofmann, Helmut Lang, Alexander Schmitt, Maxim Sidorov, Stefan Ultes, Sergey Zablotskiy, and Kseniya Zablotskaya.

A very big special thanks goes to my long-term office colleague Florian Nothdurft who was probably one of the biggest supports one could think of during the thesis. He was both a member of my working group and a SFB colleague.

The cooperation with my SFB colleagues was always friendly and helpful. Personally, I would like to thank Pascal Bercher, Thomas Geier, Thilo Hörnle, Frank Honold, Peter Kurzok, Felix Richter, Bernd Schattenberg, and Felix Schüssel for their support and collaboration especially creating the "demo scenario one", but also the many more people, whom I had discussions with and whose names are not mentioned here.

Further thanks go to the bachelor students who contributed with their work to the test platform for emotional dialogue strategies: Omneya Abdelmaksoud Rabie, May Amr Mohamed Fouad Abbas, and Mina Nabil Hanna. Additionally, I would like to thank Melina Klepsch, Kathrin Kranz, and Anja Schöllhorn for their contribution in the evaluation of the emotional dialogue strategies study. Thank you, Olesia Kukin, for your contribution to the evaluation of the adaptive dialogue system.

Now I would like to come to some more personal thank yous.

Above all, I would like to thank my girlfriend Natalie Christov, who supported and motivated me throughout the thesis and who was very understanding even if the last bus from the university home had already been missed.

Furthermore, I would like to say thank you to my parents, Anni Bertrand and Franz Bertrand, who laid the basement for even being able to begin with the thesis because they facilitated the neccessary education and also supported me during the studies and the phd thesis. Further thanks go to my sister Melanie for her help and support.

Last but not least I would like to thank the people whom I have forgotten here and who supported my work.

Contents

List of Figures **xi**

List of Tables **xiii**

1 Introduction **1**

1.1 Dialogue System Architecture 3

1.2 Adaptivity 5

1.2.1 Context Information in a Spoken Language Dialogue System . . 7

1.2.2 Context Information in a Multimodal Dialogue System 9

1.2.3 Emotional Adaptivity 12

1.3 Adaptivity in Dialogue Management 13

1.3.1 Prompt Strategies 13

1.3.2 Confirmation Strategies 13

1.3.3 Form of Initiative 14

1.3.4 Freedom in Dialogue Order 14

1.3.5 Emotional Dialogue Strategies 14

1.4 Thesis statement 15

1.5 Thesis contributions 15

2 State of the art **17**

2.1 Context Information 17

2.1.1 User Modeling 17

2.1.2 Emotion Recognition 19

2.1.2.1 Emotion Theory 19

2.1.2.2 Emotion Recognition from Speech 21

2.1.2.3 Multimodal Emotion Recognition 24

2.2 Dialogue Management System Architectures 25
2.3 Adaptivity to Context Information 29
2.3.1 Affective Computing 29
2.3.1.1 Simulating Emotion 30
2.3.1.2 Influencing the User 30
2.3.2 Motivation Research 31
2.4 Conclusion 32

3 Emotional Dialogue Strategies **35**
3.1 Different Strategies for Reacting to User Emotion 35
3.2 Preliminary Study 37
3.2.1 Description of the Experiment 37
3.3 Follow-Up Study 39
3.3.1 Experimental Setup 40
3.3.2 Goals of the Experiment 40
3.3.3 Prior Questionnaires 41
3.3.4 System Architecture 42
3.3.5 Description of the Experiment 44
3.3.6 Significance of Results for Dialogue Modeling 46
3.3.7 Results of the Study 46
3.3.8 Psychological Explanation 50
3.3.9 Significance for Dialogue Systems 51
3.4 Realization of Adaptive Dialogue Management 52
3.4.1 Design 53
3.4.2 Model 53
3.4.3 Control 54
3.5 Conclusion 54

4 Dialogue Management Approach **55**
4.1 Dialogue Model 56
4.1.1 Structure of the Model 56
4.1.1.1 Dialogue Goals 56
4.1.1.2 Dialogue Goal Components 57
4.1.2 Realizing Adaptivity 58

4.1.2.1 Guard Conditions 60
4.1.2.2 Exit Guard Conditions 60
4.1.2.3 Dialogue Initiative 61
4.1.2.4 Choice of Dialogue Order 62
4.1.3 Dialogue Widgets 65
4.1.3.1 Widget Accumulation 71
4.1.3.2 Resource Externalization 74
4.1.4 Summary for Dialogue Model 77
4.2 Dialogue Control 77
4.2.1 Initialization 78
4.2.2 Hierarchical Dialogue Goal Processing 78
4.2.3 Constraint Programming 80
4.2.4 Algorithm Description 81
4.2.5 Summary for Dialogue Control 91
4.3 Dialogue Design 91
4.3.1 Domain Design 91
4.3.2 Adaptive Domain Design 92
4.3.3 Graphical Dialogue Domain Editor 93
4.3.4 Proof of Concept 98
4.3.5 Summary of Dialogue Design 100
4.4 Conclusion 100

5 Evaluation **103**
5.1 Portability 104
5.1.1 Companion System - Home Cinema Application 104
5.1.2 Flight Booking Application 105
5.2 User Test 105
5.2.1 Adaptivity to User Emotion 106
5.2.2 Adaptivity to User Expertise 107
5.2.3 Different Forms of Initiative 109
5.2.4 Choice of Dialogue Order 111
5.2.5 Discussion of Results 114
5.3 Conclusion 115

6 Conclusion and Future Directions **117**
6.1 Emotional Dialogue Strategies 118
6.2 Adaptive Multimodal Dialogue Manager 118
6.2.1 Dialogue Model 118
6.2.2 Dialogue Design 120
6.2.3 Dialogue Control 120
6.3 Evaluation 121
6.4 Future Directions 122

Appendices **125**

A Emotional Dialogue Strategies **127**
A.1 Utterances to Express Tankfulness towards the User 127
A.2 Utterances to Praise the User in a General Way 127
A.3 Utterances to Calm the User 128
A.4 Utterances to Motivate the User 128

B Dialogue Model **131**
B.1 Guard Language 131
B.2 Logical Statement Language 132

C Evaluation **133**
C.1 User Test 133
C.1.1 Introduction Text (translated from German) 133
C.1.2 Introduction for Emotional Rounds (translated from German) . . 133
C.1.3 Adjectives used for inducing emotions (translated from German) 134
C.1.4 Questionnaires used for determining user satisfaction (translated from German) 136
C.1.5 Results of the Questionnaires 138
C.1.5.1 T Test 138
C.1.5.2 T Test Females 138
C.1.5.3 T Test Males 139

References **141**

List of Figures

1.1 Dialogue system architecture 4
1.2 Adaptivity to the User 6
1.3 Architecture of a Spoken Dialogue System 8
1.4 Architecture of a Multimodal Dialogue System 10

2.1 Different dialogue systems 26
2.2 Architecture of the MALIN dialogue system 27
2.3 Architecture of the TRIPS dialogue system 28
2.4 OWLspeak 28

3.1 Architecture of the WoZ-software 40
3.2 Components of the WoZ-software 43
3.3 Introduction text 44
3.4 Game screenshot 45
3.5 Tripartite structure of the Dialogue Manager 52

4.1 UML Class Diagram of Dialogue Model Structure 59
4.2 Example of Guards 60
4.3 Example of Exit Guards 61
4.4 Dialogue Structure with Guards 63
4.5 Dialogue Structure with Evaluated Guard Conditions 64
4.6 Question with Confirm Dialogue Part 66
4.7 Pleasure Adaptive Dialogue Part 67
4.8 Arousal Adaptive Dialogue Part 68
4.9 Dominance Adaptive Dialogue Part 69
4.10 Expertise Adaptive Dialogue Part 70

4.11 Dialogue Part for Yes-No Answer 71
4.12 Merge Process 73
4.13 Alternative Merge Process 75
4.14 Widget for Confirmation Dialogue 76
4.15 Outline of the process from model to runtime objects 79
4.16 Example of Goals transformed to Constraints 82
4.17 Example of EMFATIC language 95
4.18 Example of ECORE editor 96
4.19 Graphical Editor 97
4.20 Files for generating Graphical Editor 99
4.21 Graphical Editor 101

5.1 Architecture of the Companion System 105
5.2 Ratings of Emotional Adaptivity 107
5.3 Ratings of Emotional Adaptivity for Females 108
5.4 Ratings of Emotional Adaptivity for Males 108
5.5 Ratings of Expertise 109
5.6 Ratings of Expertise for Males 110
5.7 Ratings of Expertise for Females 110
5.8 Ratings of Expertise 111
5.9 Ratings of Initiative for Males 112
5.10 Ratings of Initiative for Females 112
5.11 Ratings of Choice of Dialogue Order 113
5.12 Ratings of Choice of Dialogue Order for Males 113
5.13 Ratings of Choice of Dialogue Order for Females 114

List of Tables

2.1 Models of Emotion 22

3.1 Overview over the rating of the different emotional strategies. 48
3.2 Overview over the amount of probands choosing each emotional strategy. 48
3.3 Rating of the different emotional strategies split into probands who chose neutral for the last round and probands who chose one or more emotional strategies for the last round. 49

LIST OF TABLES

1

Introduction

As more and more jobs that were done by human workers are replaced by computer systems, it is increasingly common to talk to a computer e.g. on a hotline or a help system. Many people can be supported by such systems if their problems are simple enough. However, the acceptance of such systems is very low.

One reason for this could be the unnatural way in which machines react to their dialogue partner. Although computational power and technical opportunities have grown enormously over the last decades, there is still a large gap between capabilities of a human and a computer.

Especially concerning the level of cognitive processes computers seem very clumsy compared to even small children. However, cognitive abilities are essential in spoken dialogues. While neuro-science still has not understood in detail how the human brain works, neuro-informatics try to simulate cognitive and learning processes based on assumptions on how the human brain might work. Although these computations are sometimes slow and difficult to understand it is possible to understand human language in a rudimentary way.

However, speech recognition and semantic analysis haven't matured to replace human understanding yet. That is why there are still more misunderstandings and misinterpretations in human-computer communication than in inter-human communication.

Besides technical deficiencies in language understanding (speech recognition and semantic analysis) there are many information channels in inter-human communication that are not yet used to the full extent in human-computer communication. For example

there is gesture, posture and mimic, which are only interpreted to a very small extent if interpreted at all.

But even if these information channels are not available, e.g. on the phone, human-human communication is still more effective than human-machine communication. What distinguishes humans from machines in an essential manner are emotions. Human ability to sense what the communication partner is feeling, human empathy and also the ability to transmit one's emotion are lacking in machines.

Another example is context. Humans usually make a lot more use of contextual information than computers. In a dialogue that is spoken next to a loud street humans will automatically adapt their voice to the noise level. They often have more context information especially if they know the person they are speaking with. They know her or his habits, likes or dislikes and the overall situation that the person is in.

However, one concept in the design of technical systems has always been to copy biological or human design. That is why there is a trend to so called companion systems. These are systems which try to personalize their behaviour to the individual user.

For this personalisation a lot of sensor data and knowledge about the user is neccessary. Sensor data may be interpreted in various ways. Sensors like cameras may provide information about the user's facial expression, his bodypose or his movements. This information may itself be used to find out how the user feels. In fact emotional adaptivity is an important topic for companion systems. There is not only research in psychology but also in cognitive technologies in order to find out about the user's emotions.

A different aspect of personalisation is preferences and likes or dislikes that are not directly available using sensor data. This information has to be collected either explicitly or implicitly. Explicit collection means that the user configures his likes or dislikes by option menus or questionnaires that are provided by the companion. Implicit collection means that the companion analyses the interaction history and tries to guess what the user likes or dislikes.

For any companion system adaptivity to the individual user is a vital aspect. However, we seem to be only at the beginning of the development of adaptive systems. Adaptivity can take place on many different layers. However for interaction between human and computer the aspect of dialogue systems is important because of its role

as a central interface. In order to understand the fields in which adaptivity in human-computer dialogues can be improved we will first consider the general architecture of dialogue systems.

1.1 Dialogue System Architecture

A dialogue system may be divided into several parts. On the one hand there is the user who interacts with the system and on the other hand there is the dialogue system which interprets the user' s input, reacts in an appropriate manner and then presents the result of the reaction in a way that the user is able to understand. This basic subdivision can be seen in figure 1.1.

In the figure we find three areas of interest for adaptivity. The system may be adaptive in the fields of *Input Processing*, *Output Processing* and the *Dialogue Manager*.

Input Processing

When the user's input is processed the system may adapt itself for instance to the user's vocabulary or to his voice in order to increase understanding. This may be realized in the speech recognizer and in the semantic analysis component. Adaptivity on the input side may also mean to switch the input device to be more comfortable to use. This is mostly done in multimodal systems by the fusion component.

Output Processing

On the output side the form of presentation may change as well as the device that is presenting. In the case of voice output e.g. the tone of the voice, even the voice itself or the loudness of the voice may vary. This may be realized by the text-to-speech component. In the case of visual output colors, graphics, and the presentation device may change. The output modality may change, too. The changing of modalities is performed by the fission component.

Dialogue Manager

Whereas adaptivity in the fields of *Input / Output Processing* means changing the form of presentation or acceptance of information, adaptivity for the *Dialogue Manager* can mean to change what is presented itself. The whole structure and even semantics of

information presented may be changed. That is why adaptivity here can be more far-reaching because the dialogue manager can also provide presentation or input suggestions to *Input / Output Processing*.

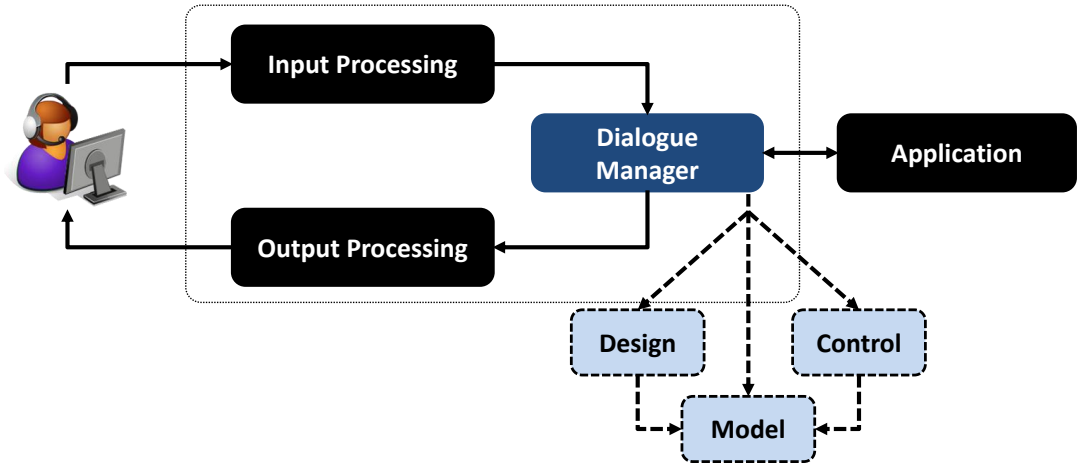

Figure 1.1: Dialogue system architecture - The figure shows different components into which a dialogue system may be divided.

Dialogue management refers to the art of interpreting the semantics filtered out by input processing and creating new semantics based on information that may be coming from an application that the user wants to operate. These newly created semantics will then be processed and output to the user in an appropriate form. The subdivision in figure 1.1 leaves room for different fields to be adaptive:

- Design
- Model
- Control

Design means the domain specific process of defining what is possible during the dialogue. It results in a *Model* that reflects all possibilities in the domain. Somehow this *Model* must be interpreted at runtime which is called dialogue *Control*.

Design

The dialogue designer usually creates a model of a certain dialogue domain. Within this domain he is free to decide how a dialogue may look like. On the one hand he has

the limit of one specific domain and on the other hand he has the freedom that the domain may provide adaptive opportunities that another domain may not.

Model

Being a frame for what the designer can do and also for what the dialogue control may handle, the dialogue model is an essential part for including adaptive features. It must be designed in a way which allows the designer to be creative when composing the domain. It should however also provide the neccessary structure for dialogue control to be able to work with it.

Control

Interpreting the dialogue model at runtime dialogue control must decide which part of the dialogue comes to execution at a certain time, depending on the user's input and on contextual information that was gathered by sensors etc.. Dialogue control can give the user a lot of freedom to guide the course of the dialogue while at the same time directing the dialogue to the goal that the user wants to achieve by using the system.

Based on the description of dialogue management it becomes clear that this field of research is especially meaningful for adaptive dialogue. It provides many possibilities to react adaptively to the contextual information that is available for the user. That is why this thesis is focused on the topic of dialogue management to make dialogue systems more adaptive.

1.2 Adaptivity

In order for a dialogue system to be adaptive the system needs to gather information about the user's context. These additional information channels may concern different factors, some of which are shown in figure 1.2.

One can see that there are many fields which may influence dialogue adaptivity. With improving sensor technology there is an increasing amount of information available that may be used to infer hypotheses about the user's context. For example, camera technology may provide information about the user's looks, his age, his gender, his emotional situation, his social status, his clothing, his background and many more. Biophysiological sensors may provide information about his blood pressure, his

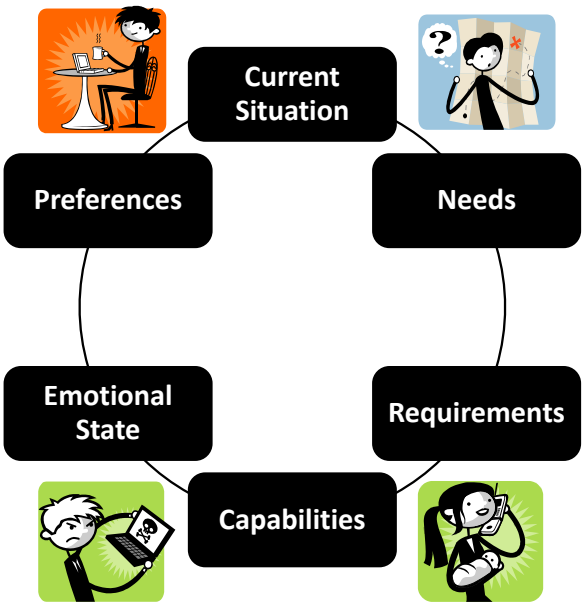

Figure 1.2: Adaptivity to the User - Users are not only different from each other but may also be in different situations that are even changing over time influencing the parameters of the current situation, needs, requirements, capabilities, emotional state and preferences.

skin conductance level, his pulse etc.. All these information may also create additional fields of interest in figure 1.2 that can influence dialogue adaptivity.

1.2.1 Context Information in a Spoken Language Dialogue System

As an increasing variety of sensors become available with evolving technology, more and more information about the user context may be collected and interpreted in order to make assumptions about the user's situation. In this light adaptivity becomes increasingly interesting for human computer interaction. Complex systems make use of dialogue system architecture and therefore the question of how to integrate adaptive features in an abstract way that leaves room for the coming developments is raised. In figure 1.2 different context parameters are shown which may influence how the dialogue should be adapted.

If we consider the architecture of a classical spoken dialogue system we find different components which may be adapted to fit the given context information. This architecture is shown in figure 1.3.

Acoustic Front-End / Speech Recognition

For speech recognition there is an acoustic front-end where the sound signal is analyzed in order to get phonemes and later words. Considering adaptivity there might be changes concerning the recognition algorithms in order to become more robust to changes that influence the speech recognizer. In the case of a very loud environment the algorithm for recognition might be altered from a more sensitive to a noise-robust one. Or there might be special recognition algorithms designed for angry users, sad users or happy users for example.

Linguistic Analysis

From the words that were recognized by the acoustic front-end semantic meanings must be filtered out in the linguistic analysis component. Here adaptivity could mean that grammars are enriched by using synonym or hyponym relationships as a means of getting meaning from words which are not found in the grammar. This approach has been used for instance by Heinroth (2012).

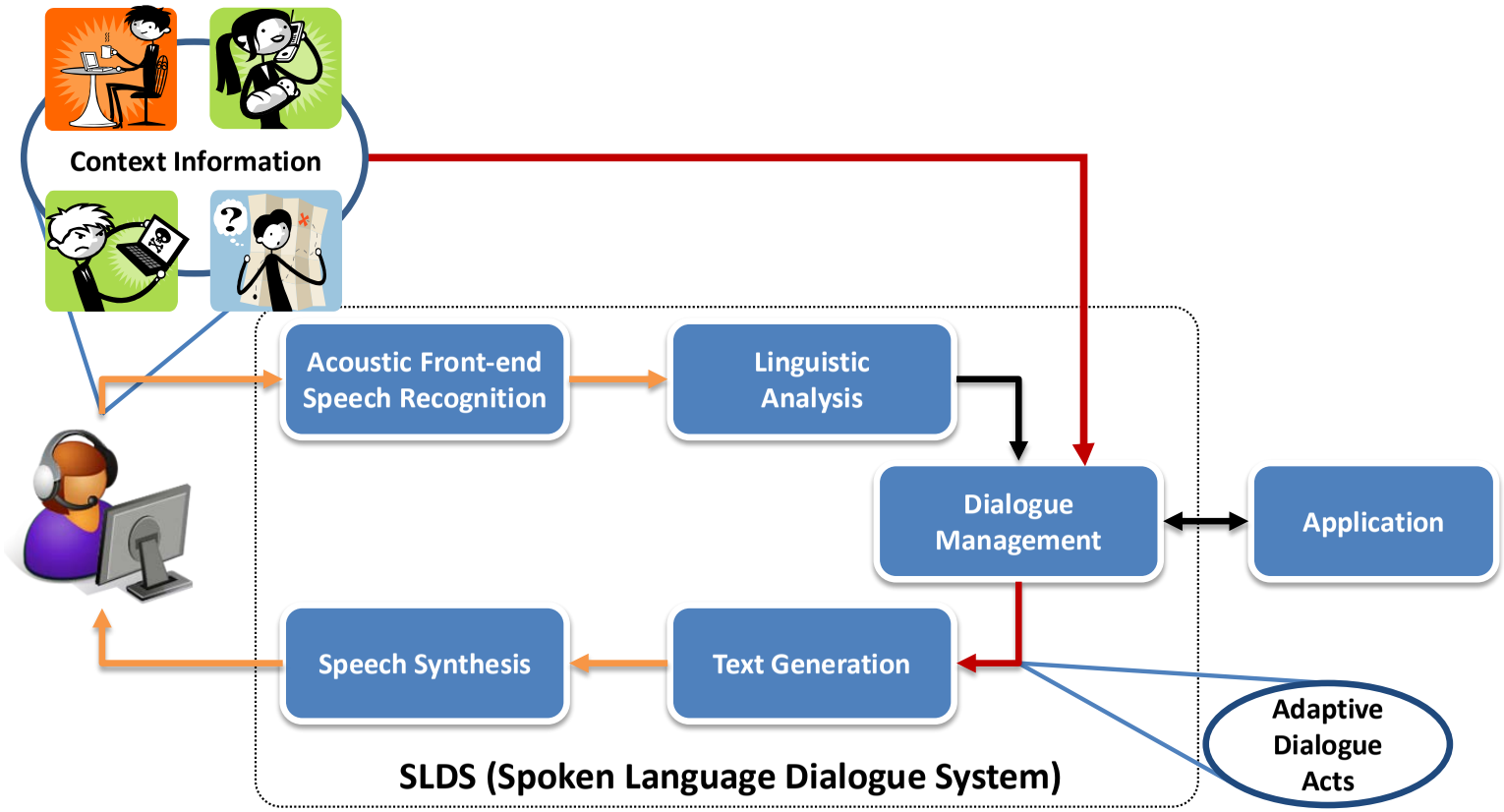

Figure 1.3: A classical dialogue system is composed of the following components: Acoustic Frontend / Speech Recognition, Linguistic Analysis, Dialogue Manager, Text Generation and Speech Synthesis. This achitecture may be complemented by context information about the user that may be provided to the Dialogue Manager which may then adapt the dialogue.

Dialogue Management

The dialogue managers decides upon what will be done next in the dialogue by analyzing the semantic meaning that is received from the linguistic analysis component. This may mean to communicate with an application, to ask further information from the user or to prepare an answer based on the data gathered from the user and / or the application. The dialogue manager then provides a semantic structure that is the result of its management process to the text generation. Adaptivity in the field of the dialogue management component may include almost everything since the dialogue manager is the heart of every dialogue system. It may vary the dialogue flow, the phrasing, and the way in which semantic information is interpreted or presented. The concrete realization of adaptivity is in most cases left to the dialogue designer who knows about the peculiarities and possibilities of the dialogue domain regarding adaptivity. In general, prompting strategies, confirmation strategies and different forms of initiative may be adjusted here.

Text Generation

Text generation will use the semantic structure provided by the dialogue management to generate text that is understandable for the user of the dialogue system. In this component adaptvity could mean to change the phrasing and the wording that corresponds to a certain semantic structure.

Speech Synthesis

In order to make the system utterances audible to the user the readable text coming from the text generation must be transformed into an acoustic signal. This is then presented to the user with the help of hardware devices. Adaptivity in this context may mean to change the voice or the tone of the voice which is used to present the given information.

1.2.2 Context Information in a Multimodal Dialogue System

As the paradigm of ubiquity finds application to dialogue systems the dialogue is split onto several modalities that the user may use to communicate with the system. In figure 1.4 a simplified architecture of a multimodal dialogue system is shown.

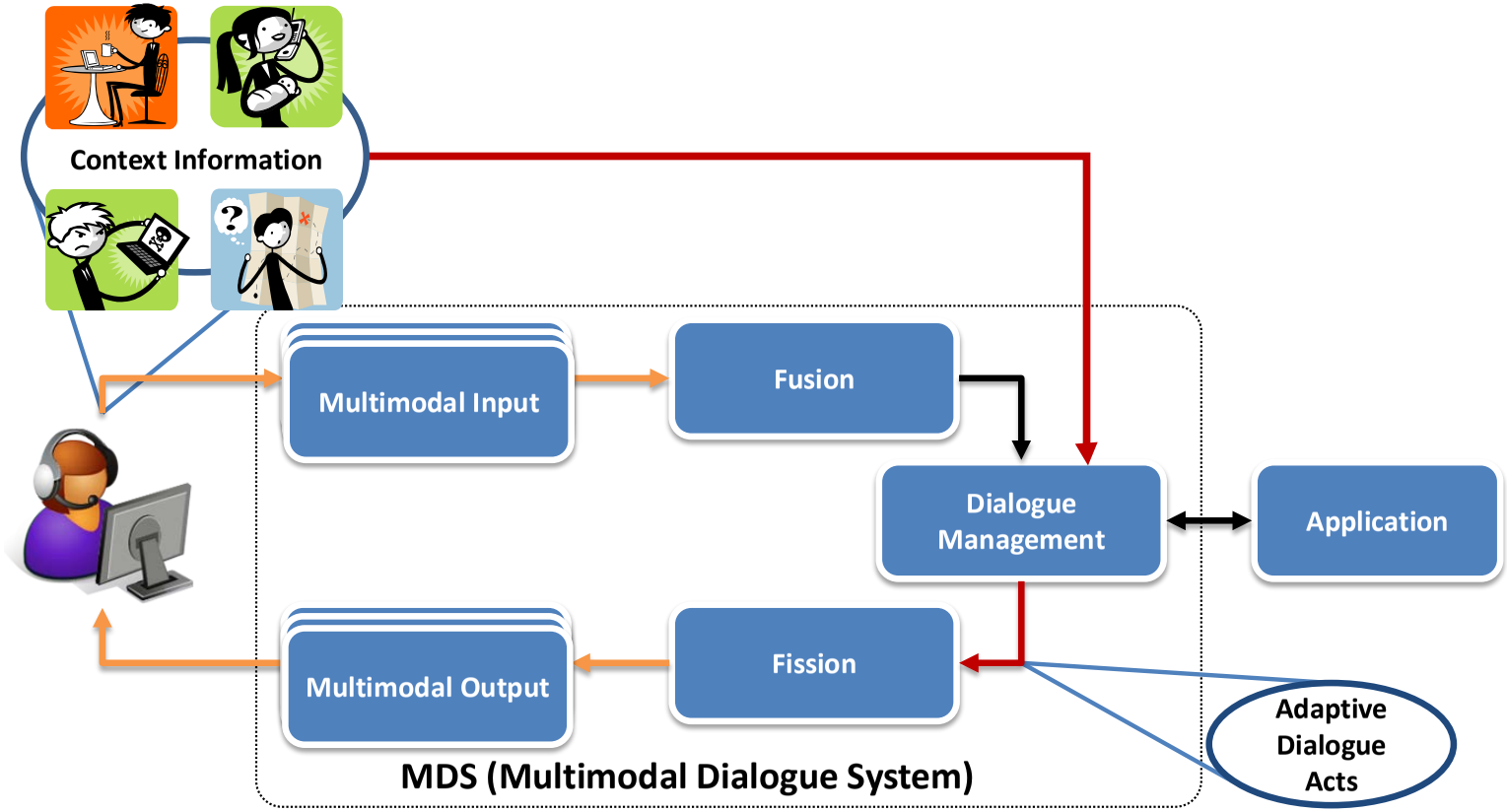

Figure 1.4: A multimodal dialogue system may be divided into the following components: Multimodal Input, Fusion, Dialogue Manager, Fission, and Multimodal Output. Context information may be available in order to adapt the dialogue accordingly.

Multimodal Input

The user may provide input via several sources. Adaptivity in this part of the multimodal dialogue system may mean that the user's input may be interpreted differently. For example if there is a lot of background noise graphical input might be prefered to speech input. However, in an in-car environment speech input might be prefered because the user needs both hands for driving.

Fusion

Using the information gathered from various multimodal input sources the fusion module interprets the information and provides a semantic value corresponding to the user's input. Adaptivity in this module may mean to change preferences for disambiguation processes. For example, an angry user input might be ignored because the user was just hitting the system and providing no meaningful input on the graphical user interface but he uttered something meaningful which may be semantically processed.

Dialogue Management

The dialogue management component interprets semantics of the user's input and prepares corresponding output. It may prefer certain modalities depending on the information that should be provided and upon the context that the user is in. As a means of adaptivity it may have certain presentational preferences that are passed to the fission component.

Fission

The fission distributes the content to be presented to the user over certain modalities. Based on the user context and upon modality propositions it may adapt the multimodal presentation to certain output devices.

Multimodal Output

Information that is conveyed to the user may be distributed over several output channels. Adapting to the circumstances certain output devices may be preferred. In addition the output devices themselves may also adapt the form of presentation of the given information to fit the needs of the current situation.

1.2.3 Emotional Adaptivity

One field of interest that is predominantly interesting for dialogue is emotion. Emotion is a vital part of human-human interaction. The dialogue partners react to each others emotional cues by changing their tone, their phrasing, or even their topic. However in human-computer dialogue the computer lags behind in recognizing the human emotions and also in reacting to them in a human-like manner.

On the one hand emotion plays an important role in communication and on the other hand the detection of emotion is relatively easy compared to detecting e.g. that the user's favorite color is green (if not asking explicitly).

Emotional Adaptivity may take different forms. In some situations it may be enough to say something encouraging to a disappointed user. In other situations there might be some error correction needed and sometimes it might be better to ignore the user's emotions and just go on with the dialogue. In all cases the process of adapting dialogue to emotion may be separated into two parts:

- Emotion Recognition
- Dialogue Adaptation

Emotion recognition is dependent on various models of emotion that are sometimes complementary and sometimes conflicting. These models stem from psychological empirical research and sometimes have a philosophical background. Sometimes these models also try to include the interaction of the user with his environment and his reactions. Other models are based on neuroscience and measured brain activities. Another aspect in emotion recognition is what kind of information channels are analyzed in order to detect emotion. Dialogue adaptation is dependent on emotion recognition and therefore may also use different emotional models in order to decide which kind of adaptation may be appropriate for the current situation. Another important aspect is the domain. The domain itself may guide which adaptive actions may be taken (e.g. a tired user may be offered a coffee if he is interacting with an intelligent environment in which a coffee machine is present).

Despite the variety of possibilities to be adaptive the dialogue system must still provide the corresponding mechanisms that allow adaptive behaviour of the dialogue at runtime. We could see that dialogue systems offer a large set of possibilities to be

adaptive in many components. However, the core of any dialogue system is the dialogue management component which also allows the most adaptivity on the structural level of the dialogue. Furthermore, many adaptations in other components of the dialogue system may be steered by the dialogue manager. That is why in the remainder of the thesis we focus on the dialogue manager as the central component for adaptivity.

1.3 Adaptivity in Dialogue Management

In order to make dialogue adaptive certain strategies have evolved. In the area of spoken language dialogue systems these are especially prompting strategies, confirmation strategies, and the form of initiative. In addition the order of the dialogue may be altered to the user's wishes.

1.3.1 Prompt Strategies

Varying the prompt strategy refers to changing the way in which information is requested from the user. For example, the prompts may be formulated as open questions allowing the user a lot of freedom in his answers to the question. However, this also means that the dialogue system needs to be able to work with the input that the user may provide to a more open question. That means an additional burden for the recognizing components and the linguistic analysis of the dialogue system. It may decrease the confidence of the recognizer as well. If the question is very precise and the user should answer in a certain way the burden for the recognizer and linguistic analysis is reduced. Additionally, the confidence of the recognizer is normally increased as well.

1.3.2 Confirmation Strategies

Adapting confirmation strategies refers to requesting different forms of confirmation for the information specified during the dialogue. In general, explicit, implicit, and no confirmation can be distinguished. Explicit confirmation means repeating the information which was given by the user and asking the user to verify if what was understood is correct. The user can answer with a simple yes or no. If he answers yes the dialogue will be able to go on. However, if the system did not understand correctly a correction dialogue must be initiated in order to provide the right information. Implicit confirmation means that the system will repeat what was understood and continue with the

next dialogue step. If the user finds that the system did not understand him correctly he has to protest and a correction dialogue will begin. No confirmation means that the system is confident that there was no misunderstanding and will not demand any confirmation from the user.

1.3.3 Form of Initiative

Adapting the form of initiative means that either the initiative lies on the system side or that the user holds the initiative. If the system has the initiative it will present system utterances that will leave the user a clearly defined place in the flow of the dialogue to answer. However if the system operates in user-initiative mode the user may choose how to begin and how to guide the dialogue with the system. Additionally there is also the possibility to lead mixed-initiative dialogues where on the one hand the system takes the initiative in guiding the dialogue but where the user may also interrupt the system in order to provide information.

1.3.4 Freedom in Dialogue Order

For providing even more adaptivity to the user it is thinkable to let the user decide what part of the dialogue he wants to start with and what part should be next instead of presenting the dialogue in a fixed order. In order to realize this form of freedom the dialogue system must not disregard dependencies in information structure of the dialogue. It may not in all cases be possible to arrange the dialogue steps in an arbitrary order specified by the users. Semantic and logical constraints in the dialogue must be modeled and respected accordingly.

1.3.5 Emotional Dialogue Strategies

To simulate an emotional dimension that a technical system does not have, utterances may be inserted into the dialogue that deal with the user's emotion. For example the dialogue system may apply thankful, praising, motivating or calming statements in order to address the user's emotional level.

Based on the introduction to the topic of adaptivity in dialogue systems we may now formulate the research of this thesis in a brief form.

1.4 Thesis statement

In short, this thesis investigates the chance arising from more and more contextual data about the user being gathered for dialogue management to become more and more adaptive to the individual user.

1.5 Thesis contributions

In order to make the dialogue adaptive to user context factors, there have been several advancements in the course of this thesis. The thesis is focused on all relevant parts for adaptivity: modelling, dialogue control, design, and more specifically on how the system may adapt to user emotion. The thesis provides a novel architecture for integrating contextual information into the dialogue. The dialogue designer as well as the algorithmic interpretation of the dialogue have guided the development of the system architecture. The system has been implemented in a prototypical way after having made a preevaluative study. The prototype has been adapted to two different scenarios of which one was multimodal. The multimodal scenario has been presented to an official commission for funding scientific projects which decided to continue the funding for this kind of research. The unimodal scenario has been evaluated in a user study which may guide future developments in this area.

Adaptive Multimodal Dialogue Model

The dialogue model needs to satisfy different constraints. On the one hand it has to be comfortable for the dialogue designer to use. On the other hand the dialogue control algorithm has to be able to interpret the designer's creation in a smooth and automatic way. For this purpose, this thesis presents a new adaptive multimodal dialogue model.

Dialogue Control Algorithm

Guiding the course of adaptive dialogue a new dialogue control algorithm has been developed which includes concepts from contstraint programming techniques into the field of dialogue systems.

Dialogue Design Interface

Adaptive dialogue enlarges the dialogue domain. This means the domain model may include different adaptive variants for the user to get to the desired goal. These variants have to specified to some extend by the dialogue designer. Especially structural differences in variants have to be modeled explicitly because these are specifically domain dependent. These affordances for the dialogue designer make it practical to use a graphical editor to specify the dialogue domain model.

Evaluation of Adaptive Dialogue Approach

The apprach presented above including the concepts and the findings of the Wizard of Oz study were evaluated in a system prototype that was compared with a prototype that did not contain the adaptive features.

2

State of the art

Adaptive dialogue systems require two prerequisites. Additional information about the user and methods to adapt the dialogue to the given information. Based on this assumption one can divide the field of research for adaptivity into gathering context information and investigating how to be adaptive to context conditions. In the next sections we will provide an overview on context information, dialogue management architectures to integrate context information into dialogue systems, and possible reactions to context information.

2.1 Context Information

For gathering context information there are two prominent approaches. One is to model the user preferences and dislikes in a model that may then adapt the user's interaction experience and personalize his dialogue. The other is the immediate detection of the user's emotion in order to adapt the dialogue to the situation at runtime.

2.1.1 User Modeling

Beginning user modeling may be attributed to Allen (1979); Cohen & Perrault (1979); Perrault *et al.* (1978) and Rich (1979a,b). In these earlier works where user modeling was part of the application system there was increaslingly made a distinction of the user modeling component like in (Allgayer *et al.*, 1989; Kass, 1991; Sleeman, 1985).

Then Finin & Drager (1986) published the "General User Modeling System" GUMS. This model allowed programmers to model stereotype hierarchies and prolog reasoning

about these stereotypes. Kobsa (1990) seems to be the first author to use the term "user modeling shell system" for systems which were "shells" in the sense that they left room for adapting the system to a model of the user. Van Melle (1981), Buchanan & Shortliffe (1984), and Shortliffe (1976) integrated their experiences with the MYCIN medical expert system into the EMYCIN system which was such an empty expert system to be filled with domain specific rules for making it a real expert system.

Then commercial expert systems became popular with systems like KnowledgeCraft, KEE and ART (see (Laurent *et al.*, 1984)).

In the early nineties several academic user modeling methodologies were developed like UMT (Brajnik & Tasso, 1994), BGP-MS (Kobsa, 1994; Pohl, 1999), DOPPELGÄNGER (Orwant, 1994), TAGUS (Paiva & Self, 1994) and um (Kay, 1995)

- UMT provided hierarchical stereotypes for user models, rules for user model inference and contradiction detection.
- BGP-MS represented stereotypical assumptions about user groups in first-order predicate logic and terminological logic. As a network server the system was able to cope with multiple users and multiple applications.
- DOPPELGÄNGER is also a server architecture accepting information from hardware and software sensors. Methods for generalizing and extrapolating data were put at the disposal of user model developers.
- TAGUS represented the user model as first-order formulas with meta-operators to express the type of information modeled. A stereotype hierarchy could be modeled and inference mechanisms, a truth maintenance system, and a diagnostic subsystem were provided.
- um was a toolkit for user modeling representing user knowledge, preferences, beliefs, and other characteristics in attribute-value pairs.

Additional shell systems were developed by Huang *et al.* (1991), Kono *et al.* (1994), Blank (1996), and Machado *et al.* (1999).

While these academical systems were not used very widely outside of the institutions in which they were developed some of the ideas of these systems were used in the late nineties when web personalization became important for e-commerce. Here the

advantages of client server architectures became prevalent to embedded user modeling components (Billsus & Pazzani, 2000; Fink & Kobsa, 2000). Especially dynamically calculated user group models were used in this context (Orwant, 1994; Paliouras *et al.*, 1999). This area of usage also brought up questions about security and protection (Schreck, 2003). Another field of research was comparing the users' action patterns using 'collaborative' filtering algorithms (Breese *et al.*, 1998; Herlocker *et al.*, 1999). As the user model might be accessed in a parallel way transactional consistency had to be guaranteed (Fink, 1999). However, these approaches were not so much focused on generality, expressiveness and inferential capabilities as the academic approaches had been but more on quick adaptation, extensibility and failover strategies. Especially most of these user models proved to be domain dependent. A detailled report on user modeling may also be found in (Kobsa, 2001).

User models provide the context information based on which a dialogue may be adapted. Considering the variety of different user models it is important for an adaptive dialogue system not to focus on only one kind of user model but to be abstract enough to include different kinds of user models.

2.1.2 Emotion Recognition

In order to describe the user's immediate state the field of emotion has found particular interest in research. Emotional information have been collected through several channels, e.g. speech, facial expression. We now give a short explanation on what the term emotion includes and about how emotional information of the user were acquired in different projects.

2.1.2.1 Emotion Theory

For understanding emotion recognition it seems best to first consider how emotions are defined. The word emotion stems from the french word "émouvoir" which means to stir up.

Different historical views have coined the view on emotions that current psychology has.

2600 years ago, Buddha explained that emotions arise from ignorance about the interconnectedness of things. He taught that the illusion of a self that is separate from its surroundings is the cause for feelings of attachment, aversion, pride, jealousy and

greed. He postulated at the same time that this illusion of a self could be overcome by studying the way things are, meditation, and action that corresponds to what one has learned. By getting rid of the illusion of a independently existing self and stiff ideas about the world one lives in, inherent qualities of the mind would shine forth. These qualities could be subsumed under the aspects of fearlessness, joy and active compassion (Nydahl, 2004).

However this explanation of human emotions never made its way into western psychology although there are striking similarities to the neurobiological theory of emotions of BIS/BAS from Carver & White (1994).

Plato suggested that the soul has a tripartite structure:

- cognition
- emotion
- motivation

This view has had a great influence on philosophy and psychology until nowadays. The effort to create separate systems for cognition, emotion and motivation has continued until today (Hilgard, 1980). Aristotle argued 50 years later for the impossibility of this separation and assumed interaction between the different levels of psychological functioning (Fortenbaugh, 1975).

Descartes inspired the philosophical and psychological debate on emotion by insisting on dealing with mental and emotional processes at once. In the 19th century Darwin (1972) was convinced that emotions served a purpose for humans in communication and survival affairs. In his view emotions evolved through natural selection and were thus universal and cross-cultural. James (1884) argued that bodily responses, rather than cognitive interpretations, are essential to emotions. According to him emotions result as interpretations of physiological states that may be triggered by external events. Cannon (1927) criticized that physiological responses were often too weak and slow to be able to explain the sometimes fast and strong subjective experience of emotions but also held the opinion that physiological factors played a big role in emotions.

These historical developments led to different categorizations of emotion and different explanations in modern psychology. Basically one may distinguish dimensional models, discrete models, meaning oriented models and componential models.

Regarding dimensional models one may differentiate unidimensional models and multimodal models. Falling into the category of unidimensional models there are models that measure valence as a scale like (Watson *et al.*, 1988) or arousal as a scale like (Duffy, 1941). A pioneer for multidimensional models was Wundt (1902). He proposed emotion to be characterized by three independent dimensions of pleasantness-unpleasantness, rest-activation, and relaxation-attention. Later Plutchik (1960) and Russell (1980) postulated two dimensional schemes where the standard emotions were aranged in a circle in the valence and arousal dimensions. Mehrabian (1996) also considered a third dimension of dominance in his famous pleasure-arousal-dominance (PAD) model of emotion.

For discrete emotion models one may distinguish circuit models and basic emotion models. Circuit models represent a neuropsychological approach to emotion and are based upon the idea of evolutionary neural circuits. Arnold (1960) and Panksepp (1982) are prominent protagonists of that tradition. Panksepp argues for four circuits producing behaviour like rage, fear, expectancy, and panic. Combinations of activities in these circuits may result in blends of these emotional states. Considering basic emotion models there is the assumption that the nature of emotions are fundamental emotions like anger, fear, joy, sadness, and disgust which form the basis for other emotions. This theory is based on the assumption that emotions are a limited number of strategies which helped man to survive. Tomkins (1963a,b) adopted the ideas of Darwin (1972) claiming that emotions were phylogenetically stable *neuromotor programs*. Ekman *et al.* (1972); Izard (1994) related this theory to the universality of emotional facial expressions.

Meaning oriented models may be split into the groups of lexical models and social constructivist models. In lexical models the wisdom of language is thought to help the researcher discover the underlying structure of a psychological phenomenon. Ortony & Clore (1988) analyzed the structure of the emotion lexicon in order to find semantic implications. Social constructivist models also consider the psychobiological components of emotion but declare them to be secondary in respect to the meaning of sociocultural context (Averill, 1980; Harre, 1986).

Componential models start with assuming that emotions are caused by a cognitive evaluation of situations and events shaping the emotion as a result of the evaluation.

Theorists of this school mainly differ by the degree of granularity that the outcoming emotions have (Arnold, 1960; Lazarus, 1968; Scherer, 1982).

Models	**Major Focus**	**Elicitation Mechanism**	**Differentiation Mechanism**
Dimensional	Subjective feeling	Rarely directly addressed; rudimentary approach-avoidance definition	Degree of similarity on feeling dimensions such as valence and activation
Discrete emotion	Motor expression or adaptive behaviour patterns	Rarely directly addressed; typical situations or stimulus configurations	Phylogenetically continuous neuroanatomical circuits or motor programs
Meaning	Verbal descriptions of subjective feelings	Rarely directly addressed; cultural interpretation patterns	Socially shared, prototypical mental representations
Componential	Link between emotion-antecedent evaluation and differentiated reaction patterns	Appraisal mechanism based on a universally valid set of criteria, influenced by cultural and individual differences	Adaptive reactions in motor expression; physiological responses to appraisal results and the action tendencies generated by the results

Table 2.1: Models of Emotion - Overview of different emotion models and their characteristics taken from (Scherer, 2000).

In table 2.1 different emotional models are summarized. A more detailled description of emotional models may also be found in (Scherer, 2000) or (Pittermann *et al.*, 2009).

2.1.2.2 Emotion Recognition from Speech

One of the channels that is used most often for emotion recognition is speech. The existing appraches may be divided into several categories. Length of speech units, usage

of different features, classification approaches and other aspects like speaker types, languages and target classes may be used to differentiate work on emotion recognition.

The segments of utterances which are used for classification are usually longer speech units like turns, utterances or phrases as in (Devillers *et al.*, 2005; Fernandez & Picard, 2005; Lee & Narayanan, 2005; Litman & Forbes-Riley, 2004; Polzin & Waibel, 2000; Schuller *et al.*, 2005; Zhang *et al.*, 2003). However also smaller units of speech are considered for classification in (Batliner *et al.*, 2003a; Vlasenko *et al.*, 2008) where words are used as units. Some approaches classify long units but work on word-level features like (Litman *et al.*, 2009; Nicholas *et al.*, 2006), special phonemes like vowels or voiced consonants like (Bitouk *et al.*, 2010; Lee *et al.*, 2004). Some studies compare various units. Schuller & Rigoll (2006) examined fixed time intervals and combined features of multiple time scales. Vogt & André (2005) compared utterances, words, words in context, and fixed time intervals.

Another way to differentiate approaches to emotion recognition from speech is the form of classification algorithm used. There are variants of SVM like in (Casale *et al.*, 2008; Meudt & Schwenker, 2012; Oudeyer, 2003; Schuller, 2006; Vlasenko *et al.*, 2008; Wu *et al.*, 2008; Zhou *et al.*, 2006), bayesian classifiers like in (Barra-chicote *et al.*, 2009; Cho *et al.*, 2007; Lugger & Yang, 2008; Mower *et al.*, 2009), neural networks like in (Petrushin, 2002; Xiao *et al.*, 2010), decision trees like in (Cichosz & Slot, 2007; Sidorova, 2009), nearest neighbor classifiers like in (Hassan & Damper, 2009), linear discriminant analysis like in (Batliner *et al.*, 2003b; Lee & Narayanan, 2005; Neiberg & Elenius, 2008), or gaussian mixture models like in (Clavel *et al.*, 2007; Lugger & Yang, 2008; Neiberg & Elenius, 2008; Sethu *et al.*, 2009; Vlasenko *et al.*, 2008).

There are also dynamic classifiers like hidden markov models which are used in (Fernandez & Picard, 2003; Kwon *et al.*, 2003; Lee *et al.*, 2004; Nwe *et al.*, 2003; Polzin & Waibel, 1998; Schuller *et al.*, 2003; Wagner *et al.*, 2007; Yun & Yoo, 2009; Zhou *et al.*, 2001). Recently hierarchical classification has found increasing interest as can be seen in (Fersini *et al.*, 2012; Kim *et al.*, 2009; Lee *et al.*, 2011; Lugger & Yang, 2008). Hierarchical classification means splitting the original classification problem into several smaller problems and is usually done along dimensions in according emotional models. Another trend is using regression models on continuous emotional scales. Examples for this approach are (Giannakopoulos *et al.*, 2009; Grimm *et al.*, 2007a).

Another interesting aspect of emotion recognition is what kind of emotions are classified. For acted speech mostly the basic emotions of (Ekman *et al.*, 1972) are used. Happiness, anger, sadness and neutral are discerned in (Oudeyer, 2003; Seppänen *et al.*, 2003). Fear, disgust, boredom, or surprise are added in (Nwe *et al.*, 2003; Petrushin, 2002; Tato *et al.*, 2002; Ververidis & Kotropoulos, 2004; Vogt & André, 2005) and happiness omitted in (Polzin & Waibel, 2000).

Considering the classification of dimensional emotion classes mostly valence and arousal are classified like in (Wöllmer *et al.*, 2008) but also dominance like in (Grimm *et al.*, 2007b), involvement like in (Barbosa, 2009) or interaction like in (Batliner *et al.*, 2008). In many cases the classified emotions prove to be task dependent. For example in telephone applications the interest lies in being able to defer critical calls to a human operator like in (Burkhardt *et al.*, 2005; Schmitt *et al.*, 2010a,b). In the area of meeting recordings categories like involvement (e.g. (Wrede & Shriberg, 2003)), or agreement vs. disagreement (e.g. (Hillard *et al.*, 2003)) are of particular interest.

In the field of spoken dialogue systems emotions are particularly task-dependent comprising confidence, confusion, frustration (e.g. (Zhang *et al.*, 2003)), uncertainty (e.g. (Pon-Barry & Shieber, 2009)), politeness (e.g. (Yildirim *et al.*, 2011)), stress (e.g. (Fernandez & Picard, 2003; Sarikaya & Gowdy, 1998; Zhou *et al.*, 2001)), or interest (e.g. (Schuller & Rigoll, 2009)).

Another field of particular interest is real-time emotion recognition from speech where expected recognition rates are lower but which is more application oriented. According to (Schuller *et al.*, 2007c; You *et al.*, 2007) spontaneous speech, realistic emotions, noise and microphone conditions as well as speaker independence are the biggest challenges in this field. Another challenge is not to know which emotions will occur during interaction. Sometimes even human labellers are unable to make a majority decision on the unknown emotions as was reported by Steidl *et al.* (2009). Truong & Van Leeuwen (2007) started to classify emotions either into known categories or into an open set of unknown emotions and were able to show that this approach is transferable to new databases unused for training. The OpenEAR system is an open-source package targeted at real-time emotion recognition introduced by Eyben *et al.* (2009). Another system in this domain is EMOVOICE by Vogt *et al.* (2008).

Looking at the topic of emotion recognition from the angle of application, one can say that a popupular field is call centers where the user may be deferred to an

operator as proposed in (Batliner *et al.*, 2003b, 2004; Burkhardt *et al.*, 2005; Devillers *et al.*, 2005; Lee & Narayanan, 2005; Neiberg *et al.*, 2006; Petrushin, 2002; Schmitt *et al.*, 2010a,b). Another domain are spoken dialogue systems for games and computer-enhanced learning software like (Ai *et al.*, 2006; Chen *et al.*, 2007; Litman & Forbes-Riley, 2004; Yildirim *et al.*, 2011; Zhang *et al.*, 2003), for robots like (Austermann *et al.*, 2005; Batliner *et al.*, 2005; Cho *et al.*, 2007), or for automotive applications like (Jones & Jonsson, 2008, 2005; Schuller *et al.*, 2007b; Tischler *et al.*, 2007). Surveillance is another context for instance in order to monitor infants like in (Satoh *et al.*, 2007) or to monitor public safety (Clavel *et al.*, 2007).

The EU project SEMAINE is targeted at building a multimodal sensitive artificial listener (SAL) which reacts accordingly to user emotions as described in (Schröder *et al.*, 2008).

Emotion recognition was also used by Ijima *et al.* (2009); Wu *et al.* (2006) in addition to speech recognition in order to improve speech recognition which may be harder for emotional speech than for neutral speech.

Other applications are the automatic annotation of court transcripts as done in (Fersini *et al.*, 2012), rating of the emotional content of movies as done in (Wang & Cheong, 2006), or the automatic processing of meeting recordings as done in (Neiberg & Elenius, 2008; Wrede & Shriberg, 2003).

2.1.2.3 Multimodal Emotion Recognition

It seems sensible to make use of several sources of information in order to increase accuracy for emotion recognition. In order to facilitate this process Wagner *et al.* (2009) created a framework for multimodal emotion recognition in real-time.

Although it may also be referenced as unimodal emotion recognition using acoustic and semantic information of language is also a form of multimodal emotion recognition. Osherenko *et al.* (2009); Pittermann *et al.* (2009); Schuller *et al.* (2005) could improve performance of only acoustic classification with using linguistic features.

Fusion of audio and visual information for emotion recognition was done by several research groups. One may divide the works on spontaneous and posed expressions that were used for recognition. Posed expressions were used by Busso *et al.* (2004); Go *et al.* (2003); Hoch *et al.* (2005); Rabie *et al.* (2009); Sebe *et al.* (2006); Song *et al.* (2004); Wang & Guan (2005) Spontaneous expressions were used by Caridakis *et al.* (2006);

Castellano *et al.* (2008); Fragopanagos & Taylor (2005); Gajšek *et al.* (2009); Pal *et al.* (2006); Petridis & Pantic (2008); Schuller *et al.* (2007a); Zeng *et al.* (2007).

A detailled survey on emotion recognition may also be found in (Vogt, 2010; Zeng *et al.*, 2009).

Emotion recognition from different biosignals has been performed by Böck *et al.* (2012); Haag *et al.* (2004); Hrabal *et al.* (2012); Kim (2007); Kim & Andre (2008); Kim *et al.* (2004); Lin *et al.* (2010); Nasoz *et al.* (2004); Picard *et al.* (2001); Takahashi (2004); Tan *et al.* (2011); Walter *et al.* (2010, 2011).

One can see that emotion recognition is a hot topic in emotion related research. With increasing accuracy in recognition adaptation of systems to emotion becomes a vision that approaches feasibility. Therefore the significance of research which is focused on possible adaptation strategies grows more and more. That is why this work is partly focused on how to integrate the information which is available based on emotion recognition into adaptive dialogue. As stated for the variety of user models the variety of different emotion models requires an adaptive dialogue management architecture which is abstract enough to include different emotion models.

Another aspect for the integration of emotional adaptivity into dialogue systems is the dialogue system architecture. Therefore in the next section different architectures are presented. Context information (especially emotional context information) may be integrated into different dialogue management architectures to a different extent and with varying ease and difficulties.

2.2 Dialogue Management System Architectures

In principle the complexity of a dialogue system increases with the complexity of its purpose. According to (Allen *et al.*, 2001), figure 2.1 lists different types of dialogue systems.

Due to different application domains that a dialogue system has been designed to work in, there are also differences in the architectural design of the system and especially in its ways to model dialogues. MALIN (cf. (Dahlbäck *et al.*, 1999)) is a well known dialogue system standing for the classical type of dialogue systems' architecture. This kind of architecture is depicted in figure 2.2. There are also different agent oriented dialogue systems, most of which are based on the open agent architecture (cf. (Martin

Technique Used	Example Task	Task Complexity	Dialogue Phenomena Handled
Finite-state script	Long-distance calling	Least complex	User answers question
Frame based	Getting train arrival and departure information		User asks questions, simple clarifications by system
Sets of contexts	Travel booking agent		Shifts between predeterminated topics
Plan-based models	Kitchen design consultant		Dynamically generated topic structures, collaborative negotiation subdialogues
Agent-based models	Disaster relief management	Most complex	Different modalities (for example, planned world and actual world)

Figure 2.1: Different dialogue systems - Illustration of the complexity of dialogue system and their architectures taken from (Allen *et al.*, 2001).

et al., 1999)). For example, the WITAS dialogue system is a prominent dialogue system even featuring multi modality (cf. (Lemon *et al.*, 2001)). For the sake of being able to communicate with their mobile robots, Lemon *et al.* (2001) developed a dialogue model with an underlying tree-like structure, called dialogue move trees.

Considering multi-party dialogue systems MITRE (cf. (LuperFoy *et al.*, 1998)) stands for an architecture, where the dialogue manager acts as a controller, which means that it's the central component in the architecture. As a consequence of the dialogue manager's central role the model utilized needs to be complex enough to integrate all the components of MITRE.

An example of agenda-based dialogue modeling can be seen in (Rudnicky & Xu, 1999). Here the dialogue model combines three crucial concepts: handler, product and agenda. Handlers take care of the interaction on a small scale, the product denotes agreed information and the agenda provides structure to the conversation topics.

The Queen's Communicator presents an approach to dialogue management which may be called object-oriented (cf. (O'Neill *et al.*, 2003)). The dialogue model contains heuristic rules related to the user as well as to the database and is divided into so called frames which give structure to the conversation.

For multimodal dialogue management there exist quite sophisticated models like the PAC-Amodeus model (cf. (Nigay & Coutaz, 1995)). It makes use of the melting pot metaphor to fuse different modalities. A melting pot is a two-dimensional data structure aligning input of different modalities on a time axis with structural parts of task objects on the other axis.

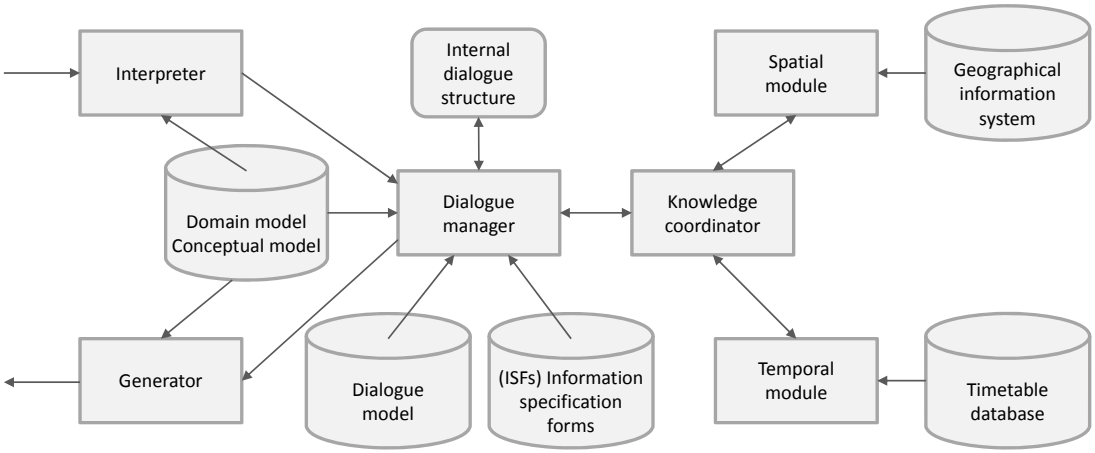

Figure 2.2: Architecture of the MALIN dialogue system - This type of architecture represents a classical dialogue system architecture. The dialogue manager plays a central role as it operates on diverse components (domain model, internal dialogue structure, knowledge coordinator, information specific forms and the dialogue model) in order to interpret input accordingly or generate appropriate output. The diagram is taken from (Dahlbäck *et al.*, 1999).

Another well-known dialogue system architecture is the TRAINS/TRIPS architecture (cf. (Allen *et al.*, 2001)) which is based on different agents coordinating the behaviour of the dialogue system. It features multimodal input which is processed by an interpretation manager working with the input based on information from different other agents. The TRIPS's architecture is depicted in figure 2.3.

A sophisticated system designed especially for the use in intelligent environments is described in (Heinroth *et al.*, 2011). Here the dialogue model is based on ontologies. One of the featured concepts is the process of on-demand generating VoiceXML in order to provide the voice user interface. Since the ontologies can be accessed and manipulated from all entities within the intelligent environment (IE) the capabilities and the configuration of the IE can influence the spoken dialogue during runtime. The basic idea is depicted in figure 2.4.

A dialogue model based on partially observable markov decision processes and dynamic decision networks was presented by Bui *et al.* (2009). This model could also include affective user states into its markov state model and split the dialogue manager into two parts: the slot-level dialogue manager and the global-level dialogue manager.

A more extensive overview over different kinds of dialogue system can be found in

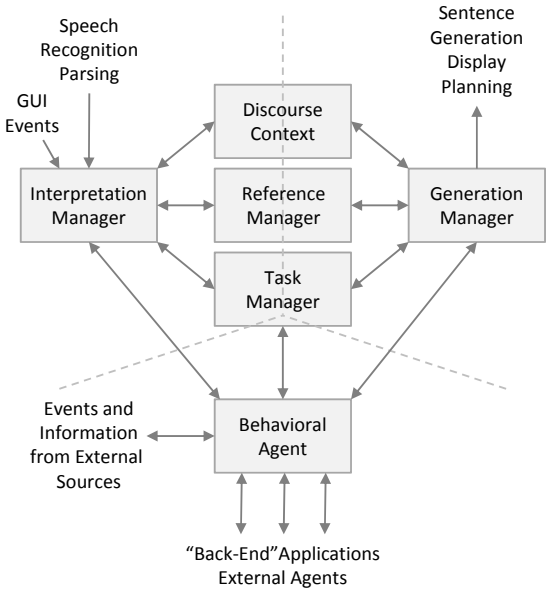

Figure 2.3: Architecture of the TRIPS dialogue system - Different agents cooperate in three different parts of the architecture: interpretation, generation and behaviour. Taken from (Allen *et al.*, 2001).

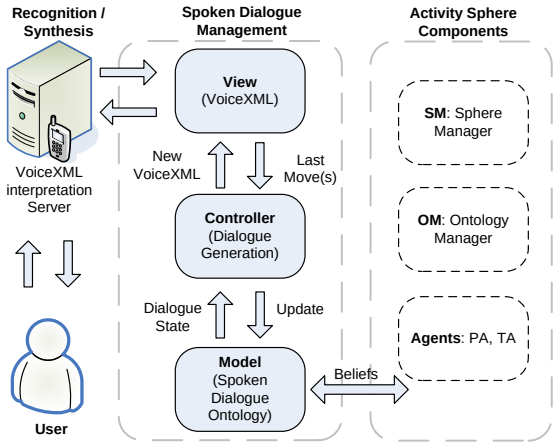

Figure 2.4: OWLspeak - The composition of the owlspeak dialogue system with model, view and controler. Taken from (Heinroth *et al.*, 2010).

(Turunen, 2004) or (Bertrand *et al.*, 2011). However none of these architectures provides a generic way to include context information and especially context information about the user's emotional state. Another problem is that many of these architectures require expert knowledge about the functioning of the way the dialogue is processed by the system which makes it difficult to model several dialogue alternatives for different context situations. In addition, either the architectures are too simple to provide the user with much freedom to influence the dialogue (in particular the temporal order) or they are too complex, so that the user won't know how to provide the correct input for the dialogue agents to react in the way he wants them to. After the consideration of different architectures in the next section research about possible reactions to context information like emotion will be discussed.

2.3 Adaptivity to Context Information

Not only incoming information are important for adaptive dialogue management, but also the form of the adaptation itself is important in the design of the dialogue management architecture. That is why the different forms of adaptation in research are discussed in this section.

In order to react appropriately to the given context different strategies have been investigated in different fields of research. On the technical side several architectures for adaptive dialogue management have been developped. On the side of research about the reactions that may be appropriate one may make a rough distinction between affective computing and motivation research.

In the field of adaptive dialogue management systems adaptivity to context may take the form of different confirmation strategies, different forms of initiative, and different prompt strategies according to McTear (2002). However more complex forms of adaptiveness like adapting the structure of the dialogue flow are also thinkable.

2.3.1 Affective Computing

For affective computing one may distinguish between approaches which try to pretend that a virtual agent has emotions, e.g. shows appraisal for user actions, and approaches which try to influence the emotions of the user, e.g. reducing frustration. The first way puts the system into the position of an artificial character that has emotions itself

while the second way concentrates on the user's character in order to affect his emotional situation. As people tend to treat computer systems like real people (see (Reeves & Nass, 1996)) it seems reasonable to create an agent architecture which simulates this behaviour. However, non-agent architectures also successfully influenced the user's emotional state.

2.3.1.1 Simulating Emotion

Regarding politeness Brown *et al.* (1987) found out universal rules which are used in language. For simulating different agent characteristics Walker *et al.* (1997) developed four strategies which are used according to a model of social interaction. These strategies for communication are direct, approval oriented, autonomy oriented or off-record. Later this model for linguistic style and the universal politeness rules were adapted by André *et al.* (2004) for spoken dialogue systems. The dialogue system is regarded as a virtual agent that tries to obey politeness rules modeled by face threat parameters.

In order to create human-like emotional agents Gratch & Marsella (2004) developed a computational framework based on a model of psychological appraisal and coping theories. The model used goals, beliefs, causal relations, plans and intention in order to simulate appraisal and coping as a result. Creating a reflexive agent with feelings De Carolis *et al.* (2001) used a planning approach in order to generate feelings and to decide on whether to show or to hide them. A quite similar approach modeling the agent's mental state on the basis of dynamic belief networks was developed in (Cavalluzzi *et al.*, 2003). In this field of emotional agents the OCC model (see (Ortony & Clore, 1988)) was widely used and Bartneck (2002) investigated the possibilities and limitations of this kind of modeling.

2.3.1.2 Influencing the User

Concerning the alleviation of user frustration Picard (2000) developped a strategy based on active listening with empathy and sympathy, on providing the possibility to amend misunderstandings, on apologizing, and social closure. Polzin & Waibel (2000) distinguished between reactions to emotion which alter dialogue prompts, feedback, and dialogue flow. Altering dialogue prompts means to provide e.g. apologetic prompts for

an angry user and short and succinct prompts for a user who is in a hurry. Changing feedback is similar to changing prompts and changing dialogue flow includes both prompts and feedback.

Using the OCC model Holzapfel & Fuegen (2002) requests domain dependent dialogue adaptation based on three layers: modifying sentence structure or grammar, modifying interaction patterns, or meta strategies containing different interaction patterns.

Active listening, empathy and sympathy were proposed as reactions to frustrated users in (Klein *et al.*, 2002) and shown to make frustrated users willing to communicate longer with the system. As control conditions ignoring the users' feelings and providing the possibility to vent the feelings were used.

In order to compare affective interventions Partala & Surakka (2004) compared positive and negative interventions after a computer delay with no intervention. They measured muscle activity of smiling and frowning muscles in the probands face. Positive interventions were found to have a significant effect on smiling muscle activity which suggests that they may be useful in human computer interaction.

Pallotta (2004) suggested to extend the information state architecture by Larsson & Traum (2000) with a mental state in order to do inference on the information provided and advocating context change as a possible system reaction to difficulties.

Embodied computer agents displaying other-oriented (empathic) emotion were compared to displaying self-oriented emotion by Brave *et al.* (2005) in user tests. Users found other-oriented emotion displaying agents more likeable and trustworthy. They perceived them more caring and felt more supported by them.

Comparing male and female embodied agents Hone (2006) could show that the influence on reducung the user's frustration was by trend higher when presented by female embodied agents and that female users by trend were more likely to be affected by empathic agent reactions than male users. The strategies used by the agents were similar to the ones presented in (Klein *et al.*, 2002).

Jaksic *et al.* (2006) could show that embodied agents may reduce moderate user frustration whereas highly frustrated users were immune to the agent's intervention or even felt increasingly frustrated.

2.3.2 Motivation Research

In the field of motivation research Schafer (1968) investigated aspects of internalization and provided a basis for other researchers like Deci & Ryan (1985) who stated that intrinsic motivation is very predominant concerning human motivation. He was partly responsible for the self determination theory of motivation. According to this theory humans strive for self determination which means that motives regarded by someone as his own are likely to be stronger than motives that someone feels to be coming from the outside. In this context Koestner *et al.* (1987) found out that praise may increase involvement and intrinsic motivation. However, Baumeister *et al.* (1990) found out that praise may also decrease the skill of the praised person who may become too self secure and make mistakes.

Another important aspect that is also involved in intrinsic motivation is gratitude which was investigated in detail in (Emmons, 2004). Investigating especially motivation in the human machine interaction context Rebolledo-Mendez *et al.* (2006) looked at the parameters confidence, independence and effort which are vital for motivation. They were able to show that a virtual character that encourages the user to exert more effort, be more confident or independent may increase the motivation of de-motivated and moderately motivated users.

Focusing on the research on empathic system reactions there have been different approaches which proved successfully that systems taking the user's emotion into account may enhance the interaction between system and user. However, these approaches mainly focused on general strategies for the system like simulating empathy. In motivation research more specific strategies have been investigated but more in context with learning and not so much focusing on the interaction between user and system. Therefore it makes sense to investigate more specific strategies in the context of user interaction with an adaptive dialogue system.

2.4 Conclusion

This overview has shown that gathering information about the user and recognizing human emotion is a topic which is highly relevant for current research in human computer interaction. There are different approaches that may help the system find assumptions about the user's general situation and his emotional state. Based on these assumptions

the system then may react in an appropriate way. Different approaches for user modeling were presented and several dialogue management systems were presented that may model complex dialogues which may also be adaptive. Considering the question of how to be adaptive especially to emotion several research areas were mentioned. However, considering the concrete form of reaction to user emotion most of the research only uses the wide term empathy or sympathy. It is still unclear which kind of reactions may be applied to emotional users and often left to domain design. In the following chapter we investigate different forms of emotional adaptivity in order to evaluate their effect on the user in a Wizard of Oz study. By doing this we may find hints about the forms of adaptivity which are desirable for dialogue systems.

3

Emotional Dialogue Strategies

Reacting to user emotions is a special form of adaptivity in the sense that the emotions occur at runtime and are often of short duration. As discussed different research studies could show the effectiveness of reacting to user emotion. In this chapter we investigate how the reaction should be in order to be effective in terms of the user's willingness to cooperate and his positive experience with the system. We present four different strategies which were investigated in two user studies.

3.1 Different Strategies for Reacting to User Emotion

Many studies about emotion in human computer interaction use the term empathy or sympathy to describe the system reaction to user frustration. The terms sympathy and empathy are quite soft terms that can be interpreted in many ways. It is often reasonable to leave the distinct kind of reaction to the domain designer as he may be able to think up concrete reactions to user emotion that are difficult to foresee before the modeling of the domain. However, it may also be reasonable to find some methods independent from the domain which might be used in dialogue systems in order to react to an emotional user. It would also be desirable that these methods are applicable in a general way. This means that they don't need a special situation to be appropriate but can be used at any time in the dialogue. For this purpose we selected four strategies:

- Expressing thankfulness towards the user
- Praising the user in a general way

- Calming the user
- Motivating the user to go on

Such strategies to influence the user's emotional state are of vital importance because even in human-human interaction understanding is not always guaranteed. When communicating with a dialogue system misunderstandings and other errors are even more relevant. Misunderstandings can cause dissatisfaction and frustration with the dialogue system. In order to remedy these shortcomings of a dialogue system we propose these strategies addressing a frustrated user.

Expressing Thankfulness towards the User

A system displaying a thankful reaction expresses appreciation of the user's interacting with the system. Thankfulness may also be expressed in context with some of the user's qualities like patience, kindness, consideration etc. that he may manifest during the dialogue.

Praising the User in a General Way

Praise shows appreciation of the user and may be used in connection with qualities that the user may disclose or that are desirable. It may direct the user's attention to his qualities and away from shortcomings of the system.

Calming the User

A user that is excited or in a hurry might be calmed by the system. A calm user naturally acts more considerate and will pay more attention to how the system may be used.

Motivating the User to go on

If a user becomes lazy or tired it is presumably best to motivate him to go on. This keeps up his willingness to cooperate and to move on in the dialogue. If the user is motivated already this strategy will be likely to have no effect on the user but won't be likely to cause any harm either.

Considering the nature of emotions being highly variable and dependent on context and circumstances, it is difficult to find any generally applicable rules. However, apart from the variability and context dependency of emotions there might be different kinds of users liking or disliking different reactions of the system to their emotions.

3.2 Preliminary Study

In order to find out which of the named strategies may be best for which kind of user we designed a Wizard of Oz experiment that compared the different strategies in a user study. A Wizard of Oz experiment is a user study with a technical system in which parts of the technical system are not yet ready to use. This means that these parts are substituted by a human who controls the system. However, the user does not know that he interacts with a technical system which is in parts controlled by a human. He thinks that the system is autonomous.

As communication is a process that is extremely dependent on emotion, it seems to be important to take into account how the user's current emotional state can be categorized. But it not only seems to be important to keep track of the user's emotional state but also to react in an appropriate way, if the user is feeling uncomfortable or even angry. Otherwise the user may abort the communication with the SLDS.

On the one hand we investigated how users differ in their reactions towards the system and on the other hand, how effective different types of strategies are. The focus of our study was on how the reaction of the dialogue system should be.

3.2.1 Description of the Experiment

Hence, we conducted an experiment, which compares how effective different support strategies are. The named strategies being thankfulness, praise, motivation and calming. The experiment was done in cooperation with the section of medical psychology at Ulm University. For control purposes we included a neutral strategy.

As a test-bed for our dialogue management system we created a Wizard of Oz (WoZ) scenario.

The following desiderata guided the scenario development:

- The user should interact with the system via spoken language in a relatively natural way.

- The system should ask the user to complete tasks that are cognitively demanding and call for attention to and engagement with the system.
- At various points the system should offer help to test the efficiency of presumably helpful dialogue strategies of the system.
- The scenario should include the induction of user emotions that are important for dialogue characteristics and occur in real-life human-computer-interactions.

We decided to implement a scenario that mainly consists of a cognitive task, where the user had to memorize visually presented items under time constraints and with increasing difficulty. All commands and answers of the user could be given via natural language. The system responded with the use of a text-to-speech engine to encourage dialogues. In order to induce emotions in a quasi-experimental way, the human operator of the system ('Wizard') could control the type of feedback the system gave (e.g. friendly or impolite), the difficulty of the task (e.g. adding countdowns to increase stress levels), the reaction time (e.g. delayed answers to cause frustration) and the quality of help (e.g. being very helpful to please the user).

With this scenario we assumed to comply with all four components mentioned above and allow the emergence of quasi-natural dialogues that form the basis of our modelling of dialogue strategies. To collect additional data and accurately assess the user's emotional state we used several psychobiological devices. A 9-channel electroencephalography (EEG) device measured brain activity over the central cortex in real-time to search for indicators of shifts in attention and emotion. Peripheral data (blood volume pulse, heart rate, skin conductance level and breathing) complemented this assessment. To measure the user's reactions (positive or negative) in the pleasure domain according to (Mehrabian, 1996) electromyography (EMG) was applied over the eye and mouth region. Additionally a frontal camera filmed facial reactions that could be assessed via automated algorithms in order to estimate the emotional content.

Subsequently a speech dialogue was initiated by the agent which requested the user to self-reflect on his performance. Due to the fact that certain steps of the experiment were designed to frustrate the user, this self-reflection was not purely positive. In response to the users answer we presented different dialogue strategies in order to find out to what extent the emotional state of the user could be influenced in a positive way.

'Positive' in this case meant that the pleasure level could be increased describing the user state according to Mehrabian (1996).

One of these strategies was randomly presented to the user by a text to speech engine simulating the SLDS. Following the user's response he was interrogated how he was feeling currently. The data acquired during the experiment was reviewed together with the user concerning his emotional state and furthermore the data was classified emotionally with the help of emotion recognition by speech, automatic EEG-analysis, skin conductivity and the analysis of the user's facial expression. Important to us was not so much what he is stating, but how he was stating it. We analyzed his statement via emotion recognition by speech and via the other available informations to investigate whether his emotional state had changed.

We had 20 probands which were tested with this system.

Unfortunately the classification of the biophysiological data was not accurate enough to compare the strategies in such a way that we had hoped. The study had shown that in order to make clear statements about the strategies compared to each other one would need direct feedback from the user. It would not be accurate enough to collect biophysical data with the intention to interpret it in order to describe the user's emotional reaction. However, this also meant that the feedback would be not as direct as biophysiological measures can be.

In order to improve on this preliminary study we designed a second study for which we also cooperated with the section of medical psychology at Ulm University.

3.3 Follow-Up Study

One particular problem when studying emotions is their empirical reproducibility. That is why, instead of trying to induce certain emotions we focused on a cognitively demanding task that ensured that the proband was involved and would in the best case be stressed.

This task was divided into several rounds in each of which the system addressed the user in a different way representing a specific emotional dialogue strategy.

Due to the fact that the reaction to the pre-defined dialogue strategies was too weak to be classified by the biophysiological data that had been recorded, the setup of the

first study was re-designed for this study now relying also on the user's feedback to the emotional dialogue strategies.

3.3.1 Experimental Setup

Owing to the different contextual and personal conditions that influence the emotions of a user, we limited our WoZ study to a setup, that has only few parameters influencing the user. We take into account the psychological tendencies of the user in a scenario designed to be as equal as possible for every user. This means that time limits, speech feedback and the timing of the feedback remained more or less the same.

As the experiment was twofold in the sense that one part was interaction with the system and the other part was emotional dialogue strategies, the tasks of interacting and providing emotional feedback are too complex for one single wizard. Thus we designed our system in a way that two wizards worked together to simulate the system. One wizard was occupied with the interaction and controls and the other wizard was occupied with the emotional dialogue strategies (cf. figure 3.1).

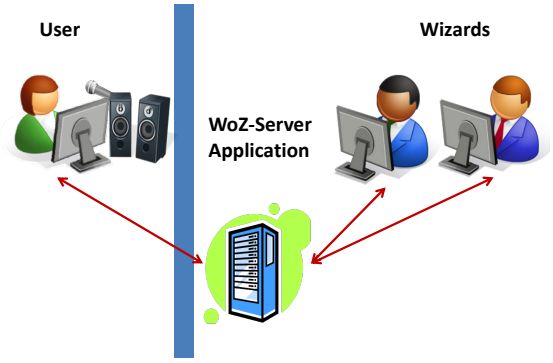

Figure 3.1: Architecture of the WoZ-software - The server architecture which provided the visual and auditory output for the user is shown. The two wizards could simultaneously control the application.

3.3.2 Goals of the Experiment

As the user cannot be put into a certain emotional state like frustration easily and repeatably, we asked him to solve a task within a limited time frame. By doing so, we simulated the usual everyday stress that everyone faces from time to time. Therefore,

this is a typical scenario for the interaction with an affective technical system. Acknowledging that the user's emotional reaction to this condition would depend on the user's character, we had the users fill out multiple questionnaires in order to distinguish them by their psychological tendencies.

The user communicated with the system via speech in order to solve the given task. The emotional feedback was effected via speech as well. The task was subdivided into a visual task that the user had to solve on the screen and into an auditory task that might interrupt the user in his concentration. The auditory task also guaranteed the user's attention to the emotional feedback because they were provided in combination requiring the user to listen.

The experiment served to distinguish between different emotional strategies and between different types of users. Thus it was partitioned into different rounds presenting the user different dialogue strategies. The aim was to find out relations between the different user types and the emotional dialogue feedback that they prefer.

The emotional feedback was divided into four categories and one neutral control feedback:

- thankfulness - *e.g. "Thank you for your focused efforts, Jane."*
- praise - *e.g. "You are doing very well, Jane."*
- calm - *e.g. "Stay relaxed, Jane."*
- motivate - *e.g. "Don't give up, Jane."*
- neutral - *nothing*

3.3.3 Prior Questionnaires

In order to be able to differentiate between the users we asked them to fill out different questionnaires before beginning with the experiment. For this purpose we used self-developed and standardized questionnaires.

Self-developed questionnaires

We used two kinds of self-developed questionniares:

- Demographic Questionnaire

- Questionnaire about the use of technical equipment

Standardized questionnaires

We used six kinds of standardized questionnaires:

- Behavioral Inhibition System (BIS)/Behavioral Approach (BAS) (cf. (Strobel *et al.*, 2001))
- Emotion Regulation Questionnaire (ERQ) (cf. (Abler & H., 2009))
- NEO-Fünf-Faktoren-Inventar [NEO-Five-Factor-Inventory] (NEO-FFI) (cf. (Borkenau & Ostendorf, 1993))
- Multi-Motiv-Gitter zu Anschluss, Leistung und Macht [The Multi-Motive-Grid of Connection, Performance and Power] (MMG) (cf. (Schmalt *et al.*, 2000))
- Experiences in Close Relationships – Revised (cf. (Ehrenthal *et al.*, 2009))
- Bern Sex-Role Inventory (cf. (Schneider-Düker & Kohler, 1988))

By analysing these questionnaires, we got an impression of the user's psychological profile.

3.3.4 System Architecture

In order to make the Wizard of Oz system work with two wizards and a proband that was located remotely certain peculiarities of this setup had to be taken into account. After consideration of all aspects we chose to make use of a servlet container and a servlet architecture in order to make the application visible as a webpage. The webpage was manipulated by one of the wizards at the speech commands of the user which were streamed via skype. The webpage was displayed for the user using teamviewer. The synchronisation between the two wizards was realized through ajax technology. In this way one wizard could see on his screen what the other had just made the text-to-speech (OpenMary) system say. In this way the wizards could coordinate the speech output of the system so that the user would not get confused. The speech data for the dialogues was fetched with hibernate from a hsql database. The different components working together are shown in figure 4.15.

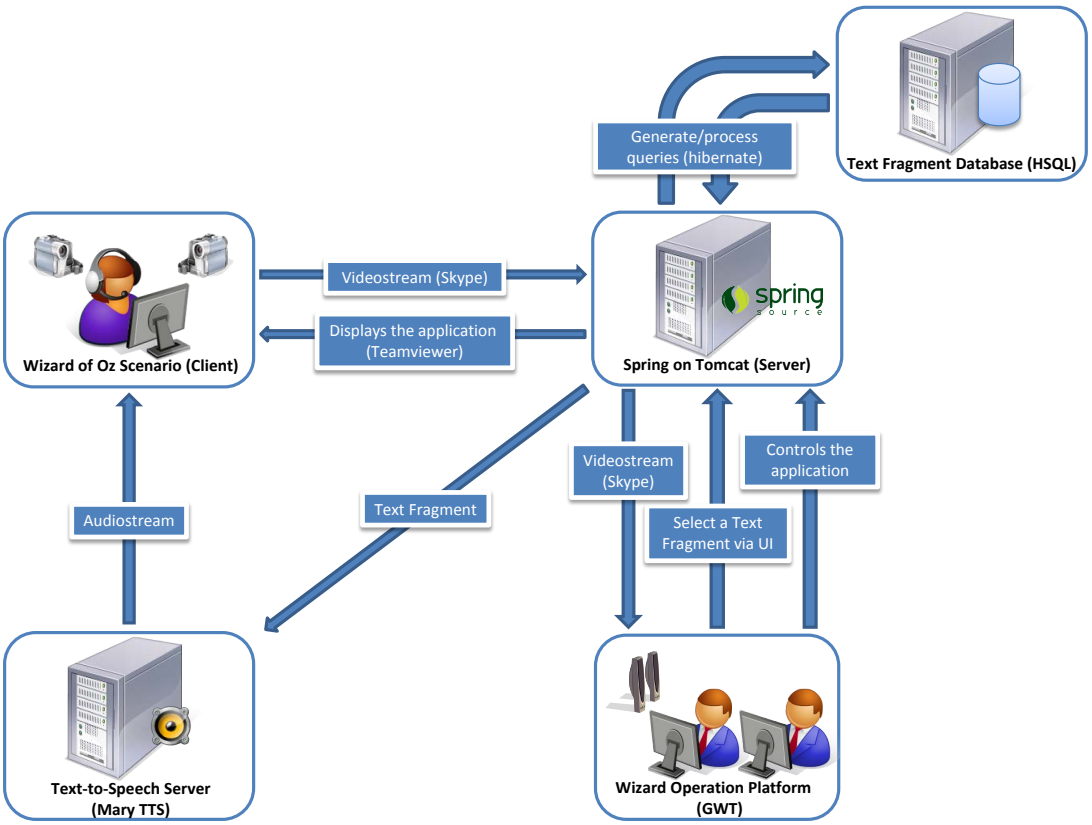

Figure 3.2: Components of the WoZ-software - The central server uses spring and hibernate to present the webapplication to the user through a webclient. The webclient is controlled by one of the wizards using teamviewer. User information are streamed via Skype so that both wizards are informed about the state and utterances of the user. For having the system read an utterance to the user the wizards may use the data stored in the hsql database or add new utterances which are then presented by the openmary text to speech system.

3.3.5 Description of the Experiment

After a short introduction to the task at hand and to the controls which are shown in figure 3.3, the user started to solve the same task in six consecutive rounds that had a maximum duration of four minutes each. The task consisted of arranging furniture in a house via speech commands that were interpreted by a robot (with the help of a wizard).

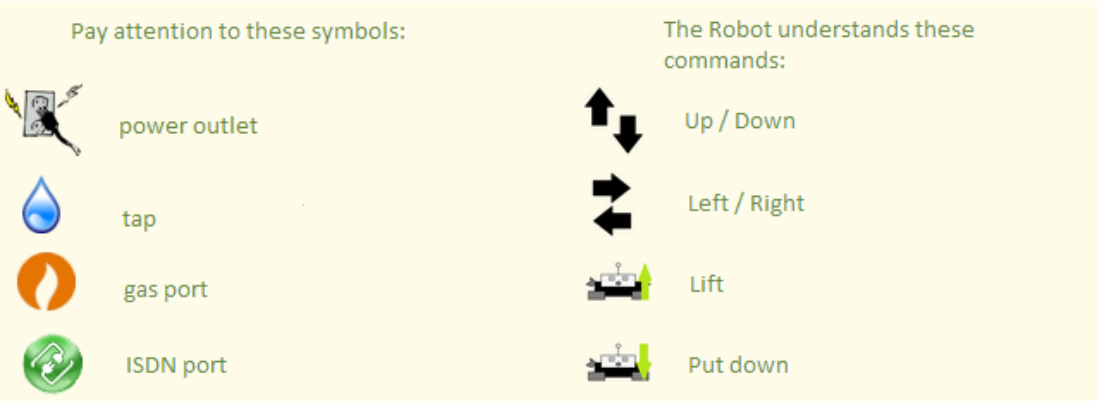

Figure 3.3: Introduction text - Before the application started the user was taught about the symbols used in the application and about the speech commands that the robot would understand.

The task was fictional and the user only saw furniture on a screen that could be moved in several directions and had to be placed into specific rooms as can be seen in figure 3.4. At the same time there were small math exercises that the user was to solve. These were presented via speech. The user got some feedback about the correct answers and each round ended either if the time was elapsed or if the user managed to arrange all the furniture.

At first, the user was introduced to the task. He was asked to arrange furniture in a given time. He could rely on the help of a robot, that is controlled via speech. At the same time he had to solve math tasks, in order to help his daughter with the homework. One wizard interpreted the user's commands to move the robot and the other wizard rendered the emotional feedback and the mathematical tasks via a text to speech software.

In each round the user was addressed in a different way. This meant that in the first round the user was addressed e.g. in a thankful manner and in the second round e.g.

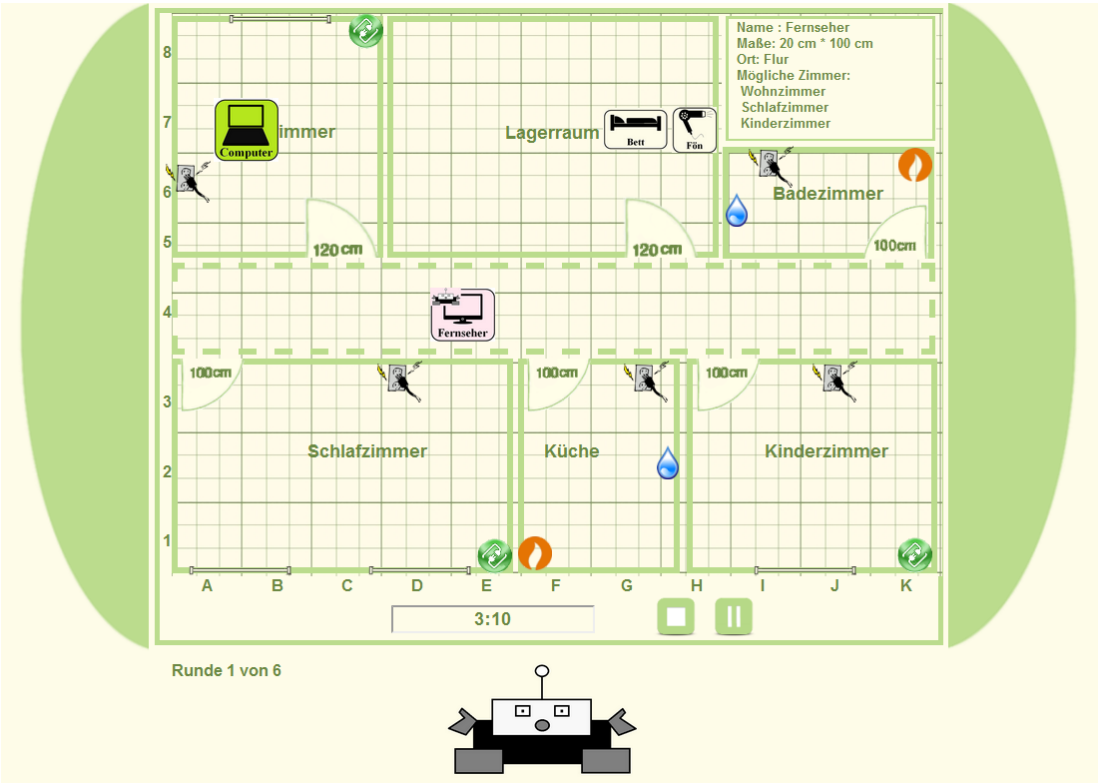

Figure 3.4: Game screenshot - The user could make the robot select furniture which was then marked red and move it up, down, to the left and to the right. He could then tell the robot to put it down again and if the furniture was placed correctly it would be marked green. All commands were given via speech and had to be interpreted by a wizard, that moved the robot. At the same time the other wizard confronted the user with the mathematical exercises and provided the emotional dialogue strategies.

in a praising manner and so on. The order of these emotional feedbacks was randomly generated to minimize effects that might arise due to their sequencing. In combination with the emotional feedback the user was presented a mathematical task to resolve. These tasks consisted of simple multiplications. Such a combination guaranteed that the user was attentive to what was said because otherwise he wouldn't be able to solve the task.

After each round the user was presented two questions after being informed if he managed to solve the task or if he failed:

- How agreeable the user found the feedback of the system.
- How he liked the way the system communicated with him.

These questions had to be answered on a 5 point likert scale (cf. (Likert, 1932)).

At the end of the fifth round the user was presented the rating he provided for the different kinds of emotional system feedback. He then chose explicitly what kinds of feedback he would like to receive from a technical system. After that, there was a final round with the user-chosen kinds of feedback given to him. Finally he rated the feedback again and by that completed the experiment.

3.3.6 Significance of Results for Dialogue Modeling

We expected to find out relations between the user's psychological tendencies that were identifiable through questionnaires and the system reaction he would prefer to his emotions. Based on these findings it would then be possible to model (emotionally) adaptive dialogues in a more user-centered way. This would mean that the system reacts to the users emotions with the ratio of different emotional dialogue strategies that corresponded to the psychological setup of the user and therefore worked best in connection with this user.

3.3.7 Results of the Study

In cooperation with the institute for education and psychology at Ulm University we investigated three hypotheses:

- The rating of the neutral dialogue strategy is better than the rating of the emotional dialogue strategies.

- The probands can be separated into several groups regarding their answers to the questionnaires, which will have different preferences for emotional dialogue strategies.

- If the emotional dialogue strategy can be chosen by the user its evaluation will be better.

After excluding 12 probands whose ratings of the system had a variance of zero we had 43 probands left. 13 were male and 30 female. The age of the probands ranged from 20 to 75 years with an average age of 36.7 years.

In order to check if gender influenced the rating of the system feedback a repeated measures anova showed no significant difference in the rating ($F < 1, n.s.$). The comparison between younger (less than 40 years) and older (more than 40 years) probands found no significant difference in rating ($F(1, 41) = 2.17, n.s.$) either.

The probands showed a significant learning effect in terms of how fast they were able to solve the tasks in the last round compared to the first round ($F(1, 36) = 112.32, p \leq .001^{***}, \eta^2 = .75$).

In order to find out if the measured personality traits had an effect on the ratings repeated measures variance analyses with the questionnaires as within subject factors were evaluated. For the measured traits (with ERQ, BIS/BAS, and NEO-FFI) was found no significant main effect for the within subject variable ($F_S < 1, n.s.$). This means that the user rating seems to be more influenced by the situation than by his personality.

Thankful, praising, motivating and calming feedback can be united as emotional dialogue strategies and compared to the neutral system behaviour. It was shown that neutral system feedback is rated better than emotional dialogue strategies ($F(1, 42) = 2.86, p_1 \leq .05^{*}, \eta^2 = .06$).

Furthermore the result of the very last round where the proband selected the strategy he liked best was compared to the mean of all other strategies. It was shown that the last round was rated significantly better than all the others ($F(1, 42) = 71.73, p \leq .001^{***}, \eta^2 = .63$).

Table 3.1 lists the ratings of each kind of emotional strategy. The rating was given on a likert scale and ranged from one to five, where five means best.

System Feedback	Mean	Standarddeviation
Neutral	3.6	0.8
Thankfulness	3.5	0.9
Motivation	3.3	1.0
Calming	3.4	0.8
Praise	3.5	0.8
Own Choice	4.1	0.6

Table 3.1: Overview over the rating of the different emotional strategies.

For the last round nearly half of the probands chose a neutral system reaction. This can be seen in table 3.2 which shows how often users chose each emotional strategy for the last round.

Chosen emotional strategy	Occurence
Neutral	20
Emotional Dialogue Strategy	23
Thankfulness	1
Motivation	4
Calming	3
Praise	2
Thankfulness and Motivation	1
Thankfulness and Calming	1
Motivation and Calming	3
Motivation and Praise	3
Calming and Praise	1
Motivation, Calming and Praise	2
Thankfulness, Motivation, Calming and Praise	2

Table 3.2: Overview over the amount of probands choosing each emotional strategy.

Probands who had chosen neutral feedback rated all emotional feedback worse ($F(1, 42) = 71.73, p \leq .001^{***}, \eta^2 = .63$), than probands who chose an emotional feedback in the last round ($F(1, 41) = 7.01, p = .05^{*}, eta^2 = .15$) which was shown with an ANOVA with emotional strategies as within subject factors and the choice of system

System Feedback	Neutral Feedback Selected		Emotional Feedback Selected	
	Mean	Standarddeviation	Mean	Standarddeviation
Neutral	3.5	0.9	3.7	0.8
Thankfulness	3.1	0.9	3.7	0.8
Motivation	3.0	1.1	3.6	0.8
Calming	3.1	0.8	3.7	0.7
Praise	3.2	0.9	3.7	0.6

Table 3.3: Rating of the different emotional strategies split into probands who chose neutral for the last round and probands who chose one or more emotional strategies for the last round.

feedback for the last round as dependent variable.

Probands who chose one or more emotional strategies for the last round didn't rate the neutral and emotional strategies significantly different. An ANOVA with within subject factors of rating of the neutral condition and rating of emotional conditions and the dependent variable of rating of the system feedback during the last round showed a significant interaction ($F(1, 41) = 5.25, p \leq .05^*, \eta^2 = .11$).

We partitioned the probands into two groups: one chose emotional dialogue strategies and the other chose the neutral strategy for the last round. It was shown that only for the neutral strategy there was no significant difference in rating while all other strategies were rated significantly better by the group that chose emotional dialogue strategies for the last round:

- neutral ($F < 1, n.s.$)
- thankful ($F(1, 41) = 5.25, p \leq .05^*, \eta^2 = .11$)
- motivating ($F(1, 41) = 3.93, p \leq .05^*, \eta^2 = .09$)
- calming ($F(1, 41) = 4.48, p \leq .05^*, \eta^2 = .10$)
- praising ($F(1, 41) = 5.96, p \leq .05^*, \eta^2 = .13$)

The corresponding means and standarddeviations are presented in table 3.3.

3.3.8 Psychological Explanation

While Partala & Surakka (2004) could show that positive feedback can lead to better performance we did not register this effect in this study. Only training effects could be seen meaning that the time needed for each round decreased during the experiment.

In order to find an explanation for the probands rating the neutral strategy better than the other strategies one can base oneself on the cognitive load theory of Chandler & Sweller (1991). Cognitive load can be divided into intrinsic, extraneous and germane cognitive load. Germane cognitive load means the effort needed to solve a task. Intrinsic cognitive load means the complexity of the task and extraneous cognitive load designates the presentation complexity of information. The emotional dialogue strategies can increase extraneous cognitive load compared to the neutral dialogue strategy. This could be an explanation why the probands prefer receiving no emotional feedback.

Unfortunately, no personality traits were identified with the questionnaires used which might predict the rating of the system feedback (ERQ, BIS/BAS and NEO-FFI). Therefore the second hypothesis could not be supported. However, probands chosing neutral system feedback rated all emotional system feedback significantly worse than probands chosing emotional system feedback for the last round. However probands who chose one or more emotional dialogue strategies for the last round didn't rate the neutral strategy significantly different from the emotional feedback. That is why it is not possible to exclude that there is a latent variable which differentiates the proband groups that we were not able to find out in this study.

The clearest result of the study was that self chosen strategies are rated best. This confirmation of the third hypothesis may be explained using the theory of cognitive dissonance or the self determination theory of motivation.

Cognitive dissonance is a state of internal tension produced by different cognitions that contradict each other. In order to weaken this state information can be interpreted or acquired selectively. Another option is to modify existing convictions that they fit to the perceived behaviour (see (Gollwitzer & Schmitt, 2009)). While doing the experiment there was no evidence that the probands liked an emotional dialogue strategy better than the others. However, after chosing one strategy they subconsciously justified their choice by rating this strategy better after the last round.

According to the self-determination theory of motivation all people strive for personal development and well-being. One basic need is autonomy in the sense of determining their own acts freely (see (Frey & Irle, 2002)). The option of being able to choose the emotional strategy freely serves this need.

3.3.9 Significance for Dialogue Systems

As it seems like emotional strategies are increasing the level of extraneous cognitive load for the user they might be useful only in cases when cognitive load is low. This means that the dialogue between user and system has to be modeled in a way that emotional dialogue strategies are only applied if cognitive load leaves room.

This could be a style guideline for the dialogue domain designer. He should not use emotional strategies too excessively in order to reduce cognitive load. The domain designer should consider the peculiarities of the dialogue domain and find spaces where it is appropriate to use emotional dialogue strategies whereas he also has to decide where the cognitive burden is too high.

The questionnaires found no indicators that could be used to infer which dialogue strategies should be used. That is why it is advisable to ask the user directly to configure the kind of emotional interaction that he prefers. This is also in line with the result of the study that the user rates self chosen emotional strategies best.

The rating of the dialogue strategies seems not to be influenced too heavily by personality traits but more by the situation context. That is why it makes sense to develop special strategies which respect the peculiarities of the dialogue domain as well as the user emotion. Being situation dependent may mean that at one point of a dialogue a completely different strategy is used than at another point of the dialogue even if the user emotion is the same.

According to the theory of cognitive dissonance the user will rate all that he can choose himself better than the same choices if they are made by the system. It is therefore vital to provide the user with the opportunity to have as much choice in the dialogue as possible and sensible.

In self-determination theory one of the key aspects of motivation is so called intrinsic motivation. For this kind of motivation autonomy is an important need as well as competence. Studies focused on choice have found out that increasing a participant's

options and choices at the same time increases their intrinsic motivation to the activities (see Zuckerman *et al.* (1978)).

For dialogue systems this signifies that these theories have important implications. Increasing the user's options and choices during the dialogue is an implication of these theories. On the one hand he will then rate the dialogue better according to the cognitive dissonance theory and on the other hand he will be more intrinsically motivated during the dialogue according to the self-determination theory of motivation.

3.4 Realization of Adaptive Dialogue Management

Having presented different areas in dialogue systems that may be adaptive we could see that dialogue management is the most influential component for integrating adaptivity. In order to realize adaptivity in this part we may consider the structure of the dialogue manager as tripartite. Dialogue management may be divided into dialogue design, dialogue model, and dialogue control. This is shown in figure 3.5.

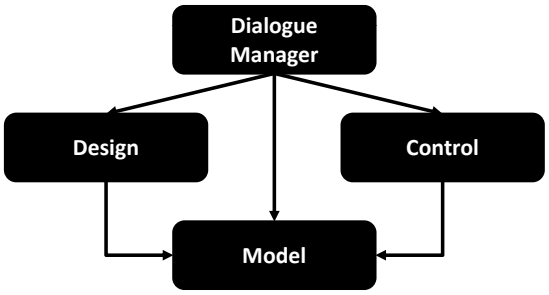

Figure 3.5: Tripartite structure of the Dialogue Manager - Dialogue Management may be seen as consisting of Design, Model, and Control. While the dialogue model specifies the framework in which the designer may work it also provides the framework for the mechanisms that control the flow of the dialogue. The model must be created before everything else as it is the foundation of design as well as control. Before runtime the dialogue design is formulated in order to specify the possible faces that the dialogue may have at runtime. Dialogue control signifies the algorithms used which select the concrete part of the dialogue model representing a form of the dialogue at runtime.

Considering this structure we discuss the requirements for adaptivity of each part of the dialogue manager.

3.4.1 Design

When designing adaptive dialogues there is a considerable additional effort to be taken compared to designing non-adaptive dialogues. This additional effort begins with a requirements analysis of the user's needs etc. considering the domain of the dialogue and the contextual situations that may influence or alter them. The designer's creativity and his ability to put himself into the user's situation are needed in order to not only model one non-adaptive dialogue but several possible variations of a dialogue that may then be selected adapting to the context at runtime. In the face of this additional effort needed the designer creates one dialogue for allowing the user to attain a specific goal several times in slightly different variations. This means that it would be handy for the designer to be able to reuse parts of the already designed dialogue to model different variations. These different variations may become confusing and the design cluttered with increasing adaptive alternatives. In order to remedy it would be handy to seperate the design into several different parts to structure the dialogue.

3.4.2 Model

The dialogue model provides the framework for dialogue design as well as for dialogue control. That means that on the one hand the dialogue designer must be able to work on the dialogue model and that on the other hand the dialogue control must be able to interpret any specified dialogue domain model and bring it to execution at runtime. On the design side we can see that an adaptive domain model is bigger than a non-adaptive model in the same domain because it needs to contain adaptive alternatives. On the dialogue control side we can see that the dialogue model needs to provide the possibility to choose at runtime between these alternatives. The dialogue model must specify the conditions under which a certain alternative should come to execution. Its task is therefore not only to represent the content and the structure of the dialogue but also the conditions which are needed to select adaptive parts. In the light of this amount of information it sounds reasonable to split the information into different parts that may then be reused separately. Separating dialogue content and structure may allow for reusability and clarity in modeling. It may also reduce redundancy in information and therefore allowing information to be reused to control the dialogue more efficiently.

3.4.3 Control

Adaptive dialogue control means to choose efficiently the optimal form of dialogue which is specified in the model. Optimal means that the dialogue corresponds to the context conditions that are measured at runtime. To be able to do this the dialogue control algorithms need to rely on context information that are available at runtime and on the other hand on conditions modeling context information in the dialogue model. The combination of context information which are available at runtime and context conditions which have been specified in the domain model by the dialogue designer or even by statistical methods may then allow to decide upon the dialogue step that fits the situation. However, any dialogue has primarily a semantic structure. Therefore, dialogue control must guide the flow of the dialogue according to the semantic structure that is specified in the dialogue model and also choose adaptive alternatives which represent the given context in the best possible way. The decision on what to do next in the dialogue relies upon the context information as well as upon semantic information that is represented in the domain model.

3.5 Conclusion

We reported on studying the user's reaction to so called emotional dialogue strategies of thankful, praising, calming or motivating utterances. As we were unable to detect any direct emotional reactions of the user we relied on his feedback for the different strategies. It could be shown that users rate the emotional dialogue strategies better if they are able to choose which ones should be applied if at all. A psychological interpretation has shown the importance of self-determination for human machine dialogue. From these results arises the paradigm to allow the user as much freedom as possible to control the dialogue. We presented how adaptivity may be integrated into spoken language dialogue systems and into multimodal dialogue systems. Giving examples of adaptive reactions of the system we concluded with a description of the integration of adaptive behaviour into the dialogue manager. We discussed how to integrate adaptive behaviour in the fields of dialogue design, dialogue model, and dialogue control. In the next chapters we will examine more closely the tasks and challenges that arise from the wish for adaptivity in these fields and we will present how they may be addressed.

4

Dialogue Management Approach

In this chapter our own tripartite approach is presented in more detail: improving the model, control, and design for enabling adaptivity in dialogue management.

This part is the central unit of our work. It combines the insights gained from the preliminary studies with the practical aspects of realizing an adaptive dialogue manager. Considerations guiding the development of the manager were multimodality, providing freedom for the user interacting with the system, and a clear structure for the dialogue designer. However, all these considerations should not only find their way into a regular dialogue manager but into an adaptive one.

To outline how this was achieved we present the different parts and structure of our dialogue model and explain how they may be used. Then we present how these parts may be used to create adaptivity, freedom for the user, dialogue widgets, and how externalisation of ressources helps for portability and multimodality. Based on this description of the dialogue model we proceed with presenting the algorithm which interprets the dialogue model at runtime and show how these aspects are brought into practice in a running system. Finally, we come to the area of dialogue design which proves to be more cost intensive in terms of time and size of the dialogue domain in an adaptive system than in an non-adaptive system. Thus we present how our dialogue model allows for providing the dialogue designer with a graphical editor to facilitate the creation of dialogue domains.

4.1 Dialogue Model

In order to allow for adaptive features like different confirmation strategies, different forms of initiative, and freedom of choosing the order of dialogue steps certain considerations in the dialogue model design have to be made. The adaptive nature requires different dialogue alternatives to be created in the domain model. These alternatives should be grouped semantically to make the domain model clear and easy to understand.

4.1.1 Structure of the Model

To give the domain a structure that is intuitively understandable we decided to rely on a hierarchical model that allows to group dialogue parts by their semantic meaning. We chose the term dialogue goal for a semantic unit because in human computer interaction there is a certain purpose why the user interacts with the system which is usually to achieve a certain goal.

4.1.1.1 Dialogue Goals

Dialogue goals are semantic units that may specify input or output steps, combine both or aggregate other dialogue goals that are semantically related. In addition, they may manipulate variables in the dialogue structure or call application logic routines. Their executability may be restricted by so called guards which specify conditions on variables. Only if these guard conditions are fulfilled the containing dialogue goal will be ready for execution.

Next Goal Relation A next goal relation may specify the goal which follows immediately in the dialogue. If there must be any external events that must occur before the dialogue can continue the current goal may contain an exit guard specifying the condition that should be fulfilled for the dialogue to continue.

Sub Goal Relation Sub goal relations represent semantic aggregations. Sub goals related to a dialogue goal mean that this dialogue goal needs them to be completed. In this way the dialogue may be structured naturally by assigning certain topical relations.

This may be realized by creating trees of goals with sub goal relations. In order to make this design more modular trees may be distributed over several files.

4.1.1.2 Dialogue Goal Components

Each dialogue goal may contain different components which specify the steps that the dialogue control will execute at runtime for the dialogue goal.

Variable Variables may be initialized using the application logic component or the effect component. They may also be referenced by guard or exit guard components in order to create conditions for the execution of dialogue goals. They may also be initialized using the input component or presented to the user using the output component.

Abstract Information Abstract information is a unit for input or output semantics that may be presented or received from the user multimodally. The semantic mapping is created as an xml file which specifies a grammar for input and rendering information for output.

Input / Output For describing if a dialogue goal should provide output or receive input the component which is named likewise may be used. This component may reference variables and abstract information components in order to specify how input may set dialogue variables or how dialogue variables may be presented.

Channel The channel component may be used to define a preferred modality by which the user is to be adressed either for output or input purposes. Visual, sound and touch are possible values for this component.

Component The component object may specify a certain component to be used for presenting system output or receiving user input.

Application Logic If the dialogue goal needs to call any application methods in order to get some information which are needed for the dialogue it may contain an application logic component. In this component the necessary information for calling the application methods are specified.

Effect The effect component works quite similar to the application logic component. However, effects specify commands for setting variables that are used internally in the dialogue.

Guard / Exit Guard In guards or exit guards conditions referring to variables are specified. There are greater than, less than, greater or equal than, less or equal than, or equal to conditions. In the case of guards the condition has to be fulfilled in order to enter the dialogue goal while in the case of exit guards the condition has to be fulfilled to finish the dialogue goal and to continue the dialogue.

Event An event is a special kind of variable which may be set externally (e.g. by an application logic call).

To give an overview over the different components and their relationships there is an UML diagram in figure 4.1.

The dialogue model may contain several dialogue goals which may be linked with goal links. These goal links indicate if the relationship between goals is a sub goal or a next goal relationship. While the dialogue goals are described through several attributes like functionalDescription and so on, they can also contain guards, exitguards application logic, effect, output and input components. Output and input components may contain component and channel components. Application logic, guard, exitguard, effect, output, and input components may link to variables (including abstract information and event components).

4.1.2 Realizing Adaptivity

Having presented the structure of the dialogue model, we may now address the question of how to implement adaptivity with it. This is the core of the dialogue manager. The idea behind the adaptive mechanism which is the foundation of our dialogue model comes from constraint programming paradigms which are also used in the dialogue control algorithm presented in section 4.2. Fundamental is the concept that different constraints influence the choice of dialogue that is best in a given situation. These constraints may be the context information gathered about the user and his situation like his emotion or his current surroundings. They are captured in so called guards.

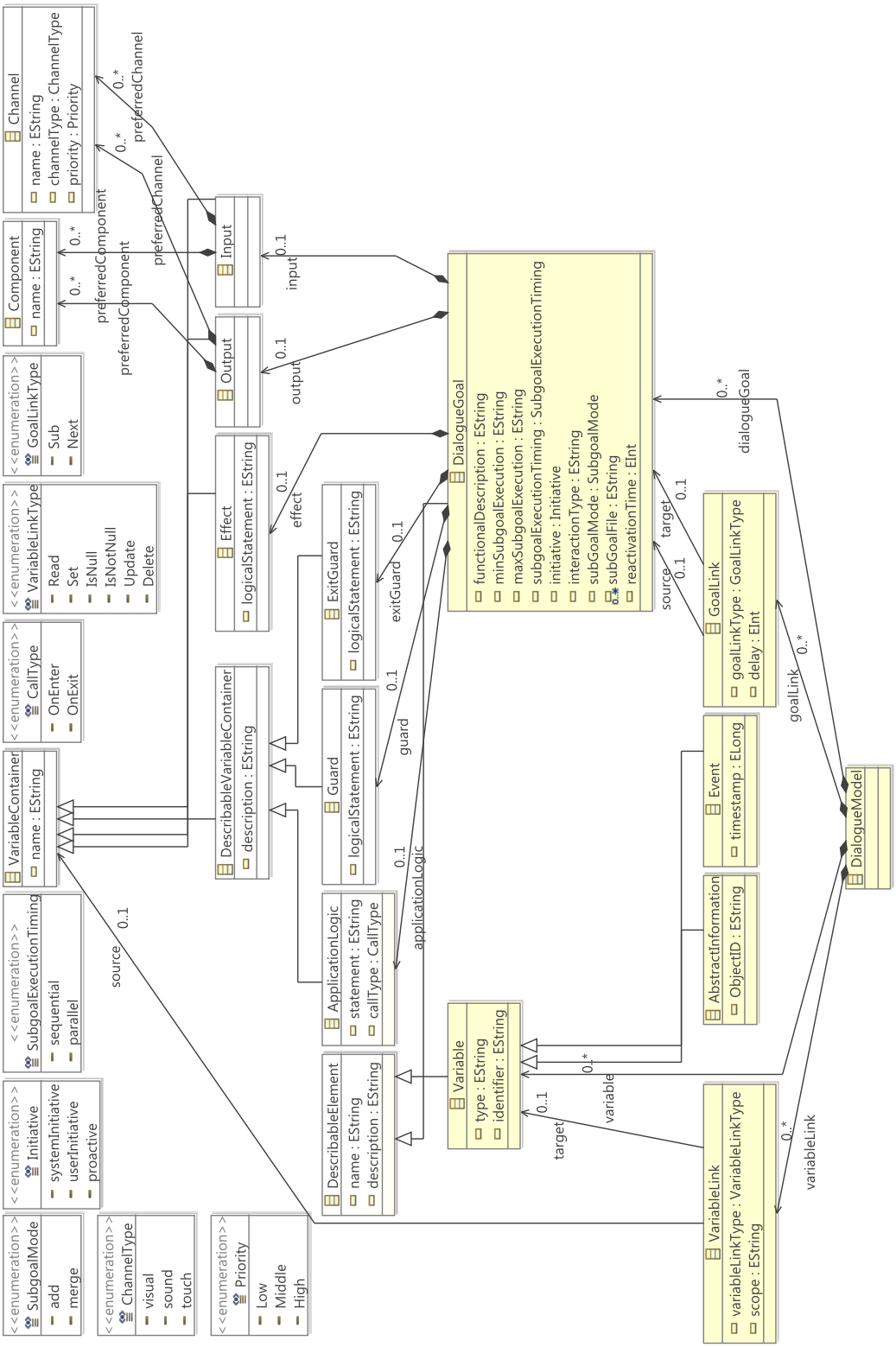

Figure 4.1: The different components which make up the dialogue model are shown in UML 2 notation.

The guard architecture allows to restrict the access to dialogue goals by creating conditions which need to be fulfilled in order to access certain parts of the dialogue.

4.1.2.1 Guard Conditions

For adapting to changing context conditions guards allow to specify conditions on variables. This means that any dialogue goal containing a guard will only be executed if the condition in the guard is fulfilled. Making use of the hierarchical structure of the dialogue model subdialogues may easily be grouped together by giving them a common parent goal. In this way dialogue alternatives which represent different versions of the dialogue adapted to context conditions may be addressed by the guard conditions. If the dialogue designer wants to create a subdialogue that should only be executed in a certain context he just needs to add a guard to the dialogue goal containing the sub dialogue. In figure 4.2 a short example of guards in a dialogue structure is shown.

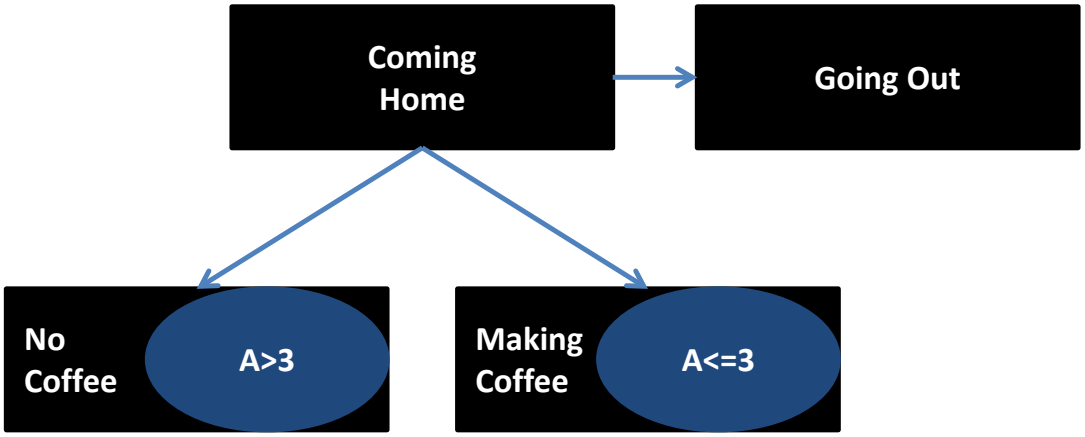

Figure 4.2: Example of Guards - Dialogue goals are displayed as rectangular shapes while guards are shown as ellipses. Arrows pointing down are sub goal relations and arrows pointing to the right are next goal relations. At runtime on entering dialogue goal *Coming Home* the value of variable A (for arousal) will decide which subdialogue may come to execution. If A (for arousal) is greater than 3 dialogue goal *No Coffee* will come to execution and if A is less than or equal to 3 dialogue goal *Making Coffee* will be executed.

4.1.2.2 Exit Guard Conditions

In contrast to guard conditions which may prevent to enter a certain dialogue goal, exit guard conditions may prevent to exit a dialogue goal. This allows to wait for events

that have to occur in the dialogue, which may be input events in multimodal dialogue or application events. In this manner dialogue continuation may be paused unti a certain event occurs.

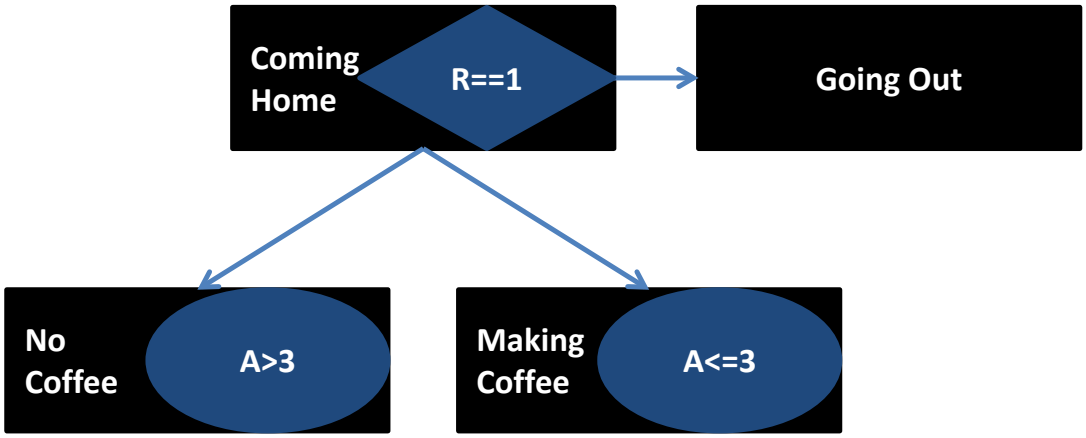

Figure 4.3: Example of Exit Guards - Dialogue goals are displayed as rectangular shapes while guards are shown as ellipses and exit guards are shown as rhombi. On entering dialogue goal *Coming Home* at runtime dialogue goals *No Coffee* or *Making Coffee* will be executed depending on the value of variable A (for arousal). However, the dialogue will not continue until the value of variable R (indicating if the user is ready) is set to 1.

Using guards and exit guards the dialogue designer may control the dialogue flow at runtime. It is advisable to design dialogue guards in a way that the dialogue will be able to continue regardless of the variable configurations at runtime, if they don't hinder the underlying logic of the dialogue.

4.1.2.3 Dialogue Initiative

Each dialogue goal may be characterized by its type of initiative. There may be user initiative, system initiative and proactive dialogue goals. This refers to their mode of execution.

User Initiative User initiative dialogue goals will be inactive until a part of their specified input is triggered (e.g. a certain word from the grammar of the user initiative goal is uttered by the user). If activated in this way the user initiative dialogue goal will interrupt the dialogue and may start a subdialogue that will resume the original dialogue after it has been finished.

Proactive Proactive dialogue goals will be inactive until their guard conditions are fulfilled. Then being activated they will initiate a proactive sub dialogue that will resume the original dialogue after finishing.

System Initiative System initiative dialogue goals are activated if there are no pending proactive or user initiative goals. However, their execution may be restricted with guard conditions.

4.1.2.4 Choice of Dialogue Order

In order to allow the user to define the order in which the different dialogue steps should be executed the dialogue structure in combination with the dialogue guards provides a generic method to decide which dialogue goals are executable. Based on the hierarchical structure there is a clearly defined set of dialogue goals active at any given time in the dialogue. Based on this set of active dialogue goals a subset of executable dialogue goals may be determined referring to the guards which constrain their executability through variable conditions to be fulfilled.

In figure 4.4 a dialogue is modeled which shows the dialogue goal *Coming Home* describing different tasks that a user may have coming home after work. They may be divided into *Sofa Activities*, *Calling A Friend*, and *Making Dinner*. Sofa Activities are divided into *Relaxing* and *Emails* depending on the measured values for P and A (reflecting the user's pleasure and arousal level). For *Calling A Friend* and *Making Dinner* certain prerequisites on the values of P and A have to be fulfilled as well.

When looking for similarities with other kinds of modeling the dialogue model may be compared to a decision tree (see (Quinlan, 1986)) but the next goal relation falls out of that category. In this comparison it may be seen as a cause-and-effect relation.

Based on the dialogue guards which control the access of the dialogue parts at runtime, adaptivity can be achieved by simple guard conditions. The dialogue parts that are accessible during runtime cannot only differ in the manner of presentation (like calming an angry user or motivating a depressed one) but also in the structure of the dialogue. A dialogue designer with a psychological background can therefore model a dialogue according to his knowledge how to cope with a user in a specific emotional situation.

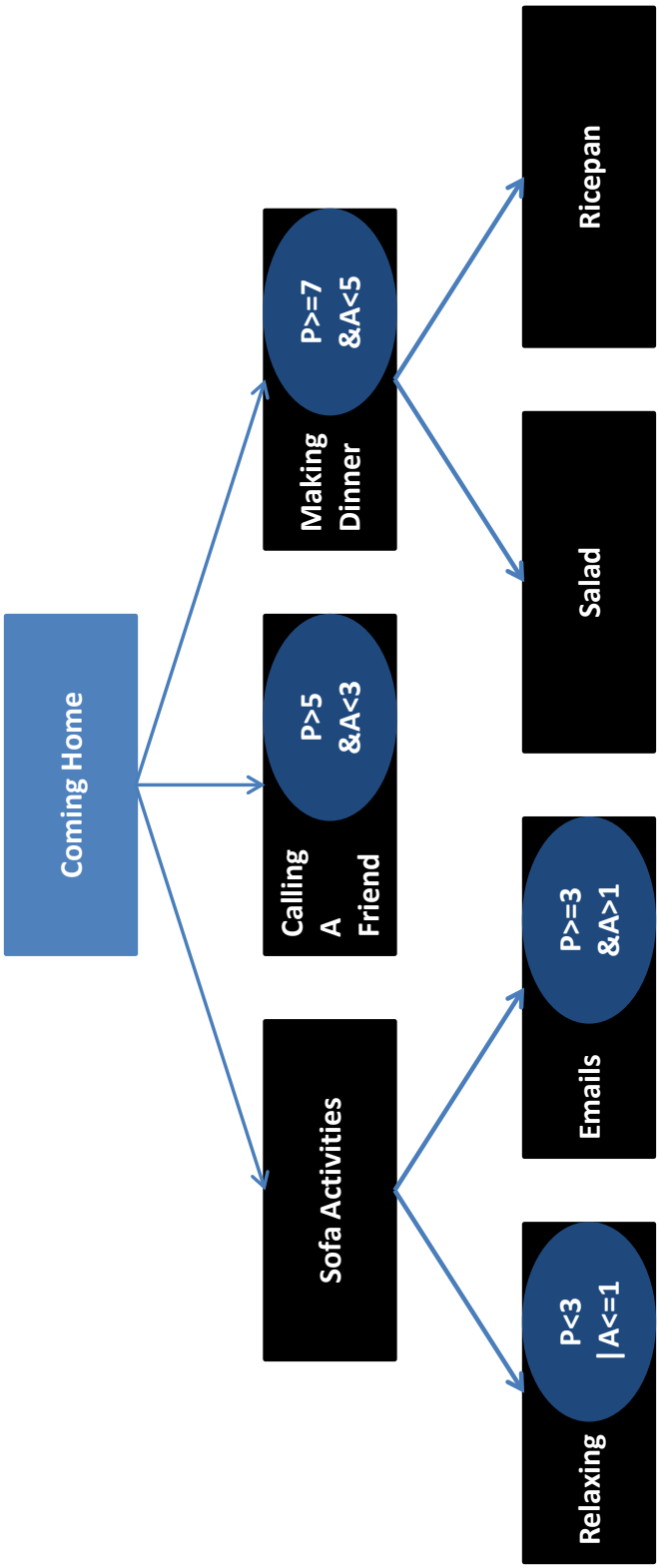

Figure 4.4: The displayed hierarchical structure contains several guard conditions which may be evaluated at runtime when the variable values at the time of the activation of the dialogue goal are known.

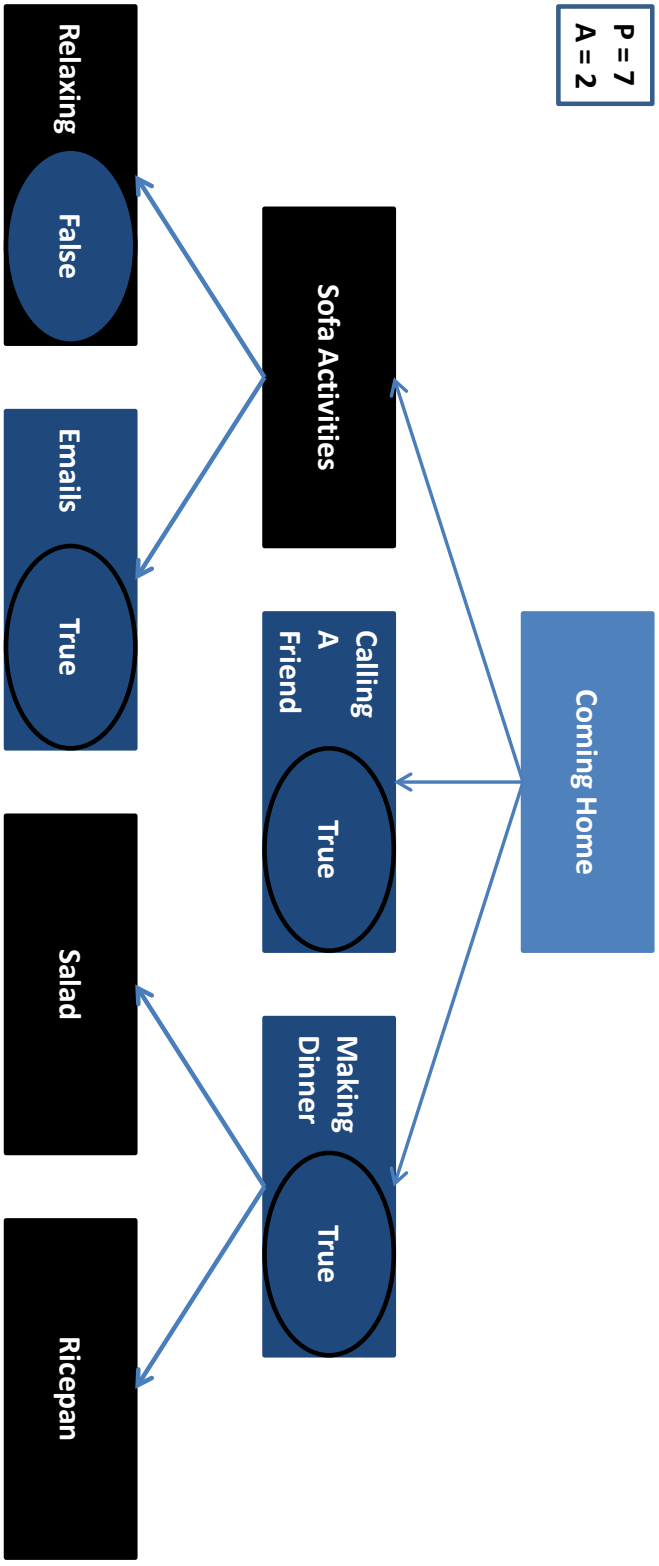

Figure 4.5: The same hierarchical dialogue structure as in figure 4.4 after the guard conditions have been evaluated leaving different dialogue goals executable.

4.1.3 Dialogue Widgets

As discussed previously adaptive dialogue means enlargement of the dialogue model compared to non-adaptive dialogue. For example a non-adaptive dialogue may just greet the user with "hello". A dialogue that adapts to the user's arousal level may greet the user with "hello, you look tired, would you like some coffee?", or "hello, you look agitated, would you like to take a rest?", or just "hello, how are you?" depending on the arousal context information that is measured by sensors.

This fact of increased dialogue model size makes reusing parts of already modeled dialogues interesting for the designer. That is why we propose the concept of dialogue widgets.

A widget is a concept that comes from graphical user interfaces and denotes reusable graphical elements that fulfill the function of allowing direct manipulation of application data. In speech dialogues as well there are parts that recur like yes-no questions, confirmation dialogues or simple data presentations. We propose to transfer the concept of widgets to dialogue systems. That means representing certain structures of information exchange between the system and the user in a way that is reusable.

In the case of our dialogue model this means that a certain amount of dialogue goals reflecting a semantic unit of the dialogue should be made reusable. This reusability is achieved in the following way.

The dialogue widget (meaning the structural unit of the dialogue just mentioned) is kept in a separate file that can then be referenced by other parts of our dialogue model. This is possible because of the hierarchical structure of our dialogue model. A certain file can be included as a sub goal because the sub file has one dialogue goal at the top of the tree like structure it contains.

This process is shown in examples of a dialogue part in figure 4.6. This sub dialogue fulfills the function of presenting a question to the user and possibly letting him confirm the input, if the certainty of the speech recognition is not high enough. The goal at the top of the sub dialogue *DGAsk* has a single sub goal *DGQuestionOutputSG* which includes files containing the adaptive output sub dialogues (shown in figures 4.7, 4.8, 4.9, and 4.10). When the output has been presented to the user he may enter his answer in *DGQuestionInput* where the abstract information container *AIQuestion* specifies the grammar which fills the variable *VQuestion* with a corresponding value. Then the user

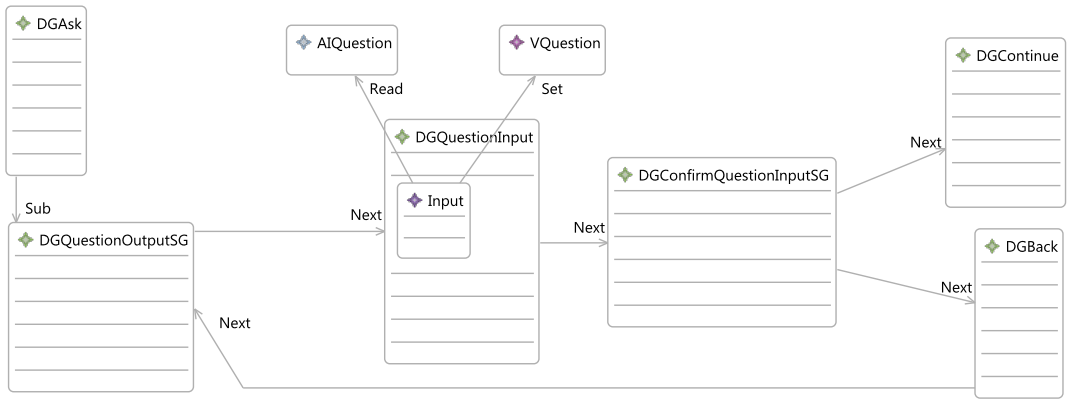

Figure 4.6: Question with Confirm Dialogue Part - A part of a dialogue as it is displayed in the graphical editor (cf. (Bertrand *et al.*, 2011).

should either confirm his answer if the speech recognizer certainty is not high enough – this sub dialogue is specified in another file which is displayed in figure 4.14 – or he will continue the dialogue with *DGContinue*. If the confirmation dialogue results in the need for a correction the user will be directed back via *DGBack* and asked again. When including the dialogue widget presented in figure 4.7 all the sub goals of *DGAdaptivePOutput* become sub goals of *DGQuestionOutputSG*. In this way the sub file is integrated into the dialogue.

In figures 4.7, 4.8, 4.9, and 4.10 one can see examples of adaptive output. The goal at the top of this sub dialogue *DGAdaptivePOutput* in figure 4.7 has three sub goals *DGHPOutput, DGLPOutput* and *DGNPNothing*. Each of these has a guard which reads the *Situation* information from the variable *VUser*. The guards specify conditions on the *Pleasure* in the *Situation* object of the variable *VUser*. If the *Pleasure* value is high, *DGHPOutput* comes to execution, if it is low, *DGLPOutput* comes to execution, and if it is neutral *DGNPNothing* comes to execution. *DGHPOutput*, and *DGLPOutput* each link to an abstract information object *AIHP*, and *AILP* which represent an output that is adapted to the user's corresponding emotion. After the adaptive output has been presented all these sub goals link to the next goal *DGNOutput* which then presents the information in an neutral way. Figures 4.8, 4.9, and 4.10 specify their output analogously.

In figure 4.11 a dialogue widget for a yes-no answer is shown. The dialogue goal *DGYesNoInput* at the top of this sub dialogue contains a single sub goal *DGYesNoInput*

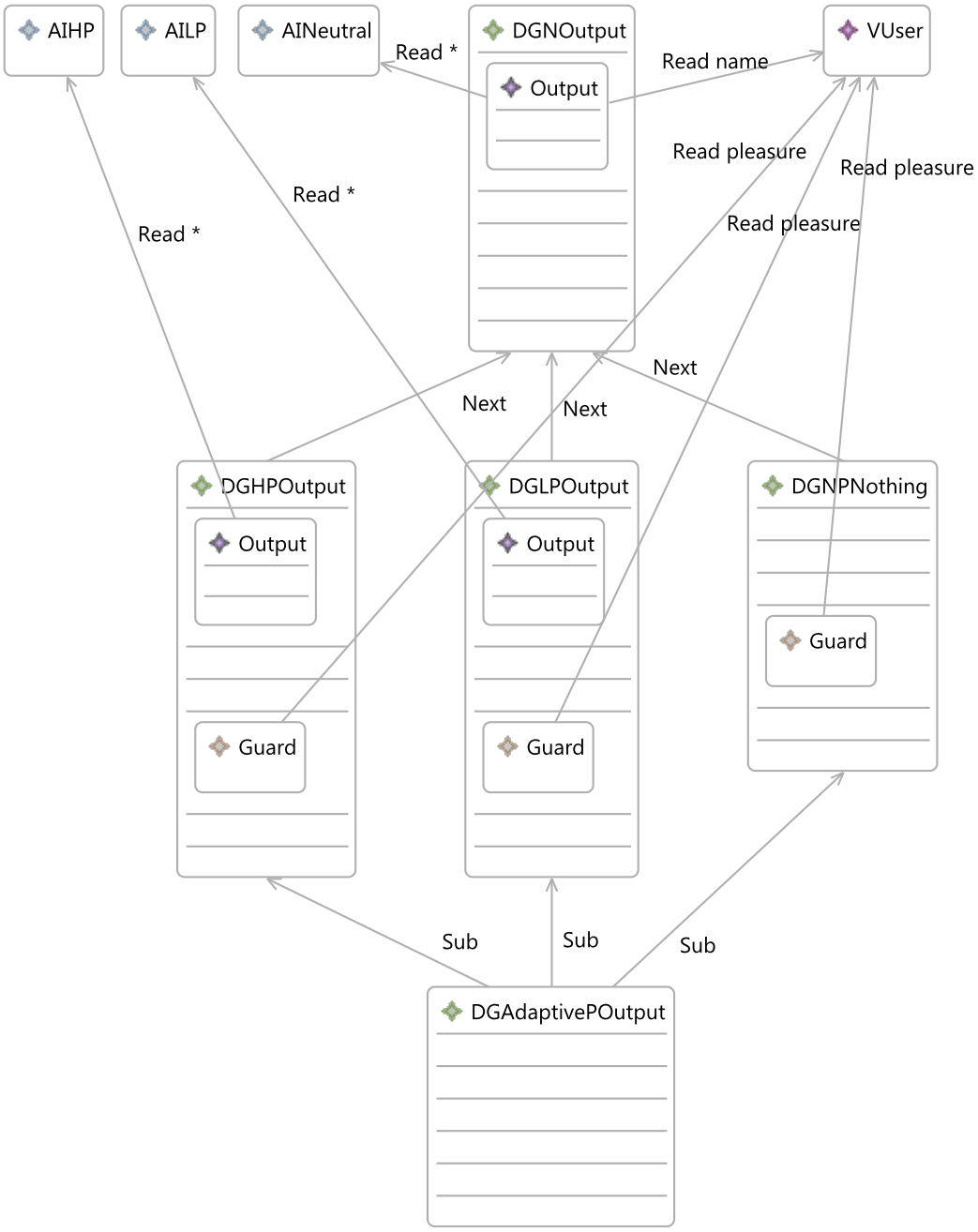

Figure 4.7: Pleasure Adaptive Dialogue Part - A part of a dialogue that presents some output to the user, which is adaptive to his pleasure level (cf. Mehrabian (1996)).

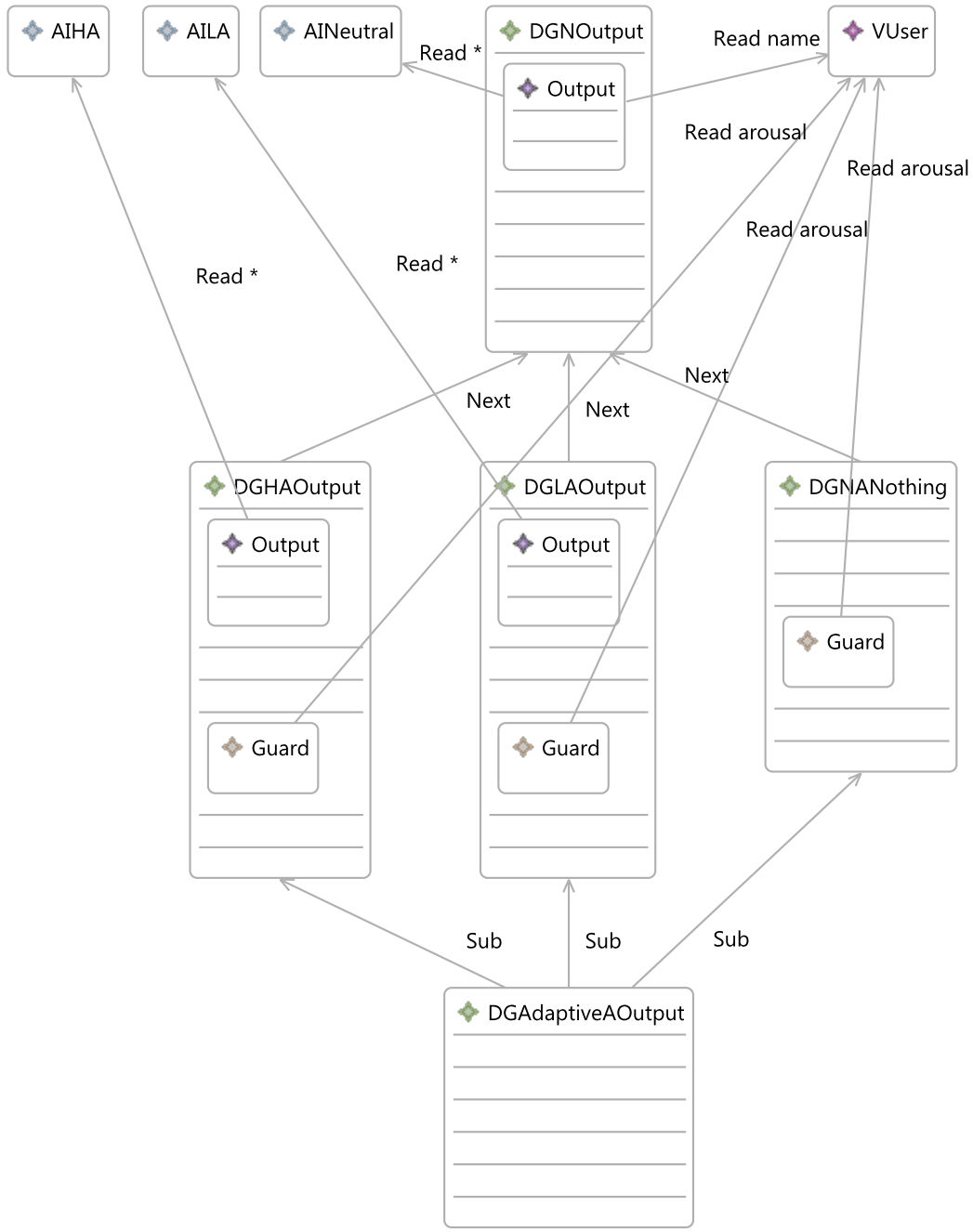

Figure 4.8: Arousal Adaptive Dialogue Part - A part of a dialogue that presents some output to the user, which is adaptive to his arousal level.

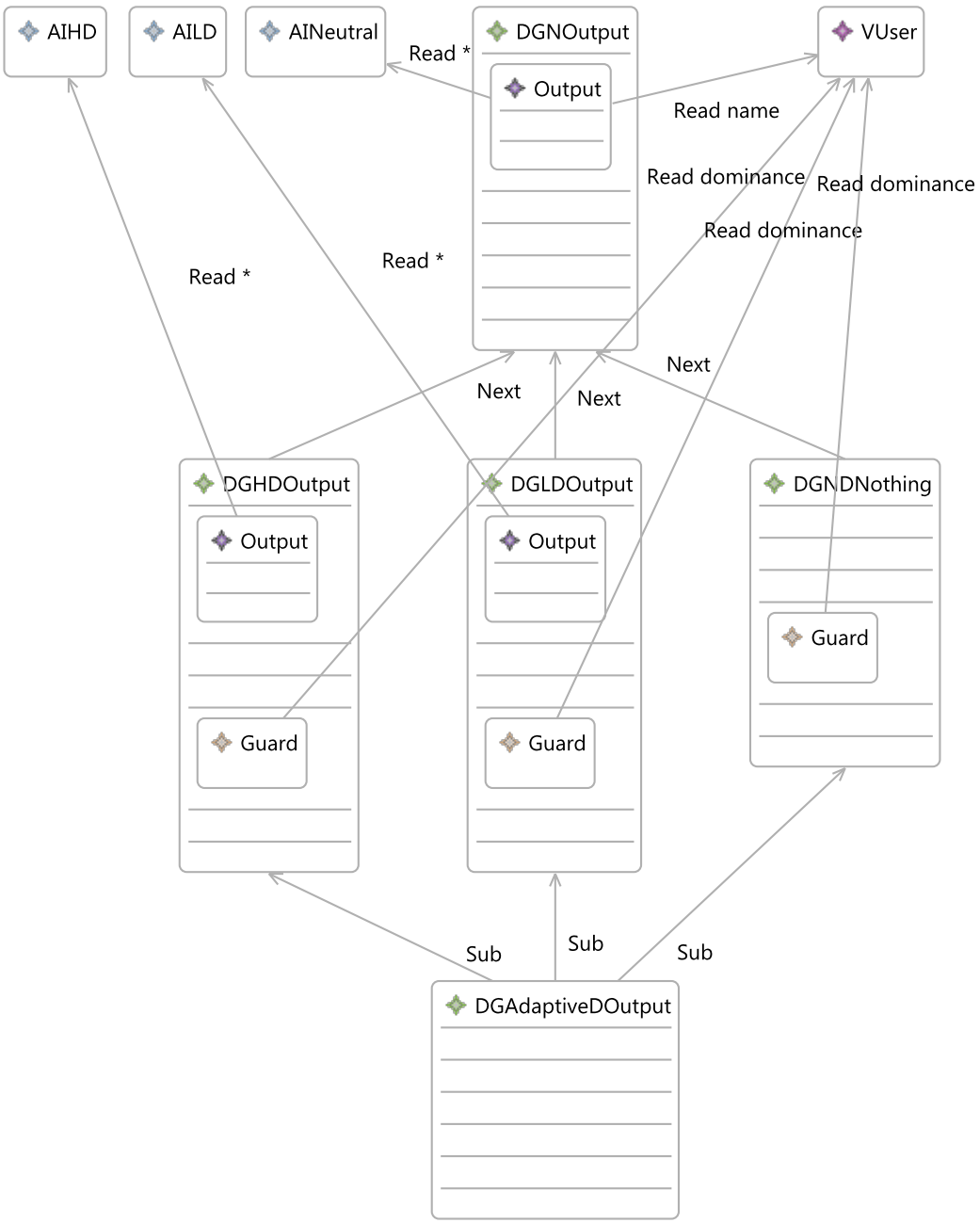

Figure 4.9: Dominance Adaptive Dialogue Part - A part of a dialogue that presents some output to the user, which is adaptive to his dominance level.

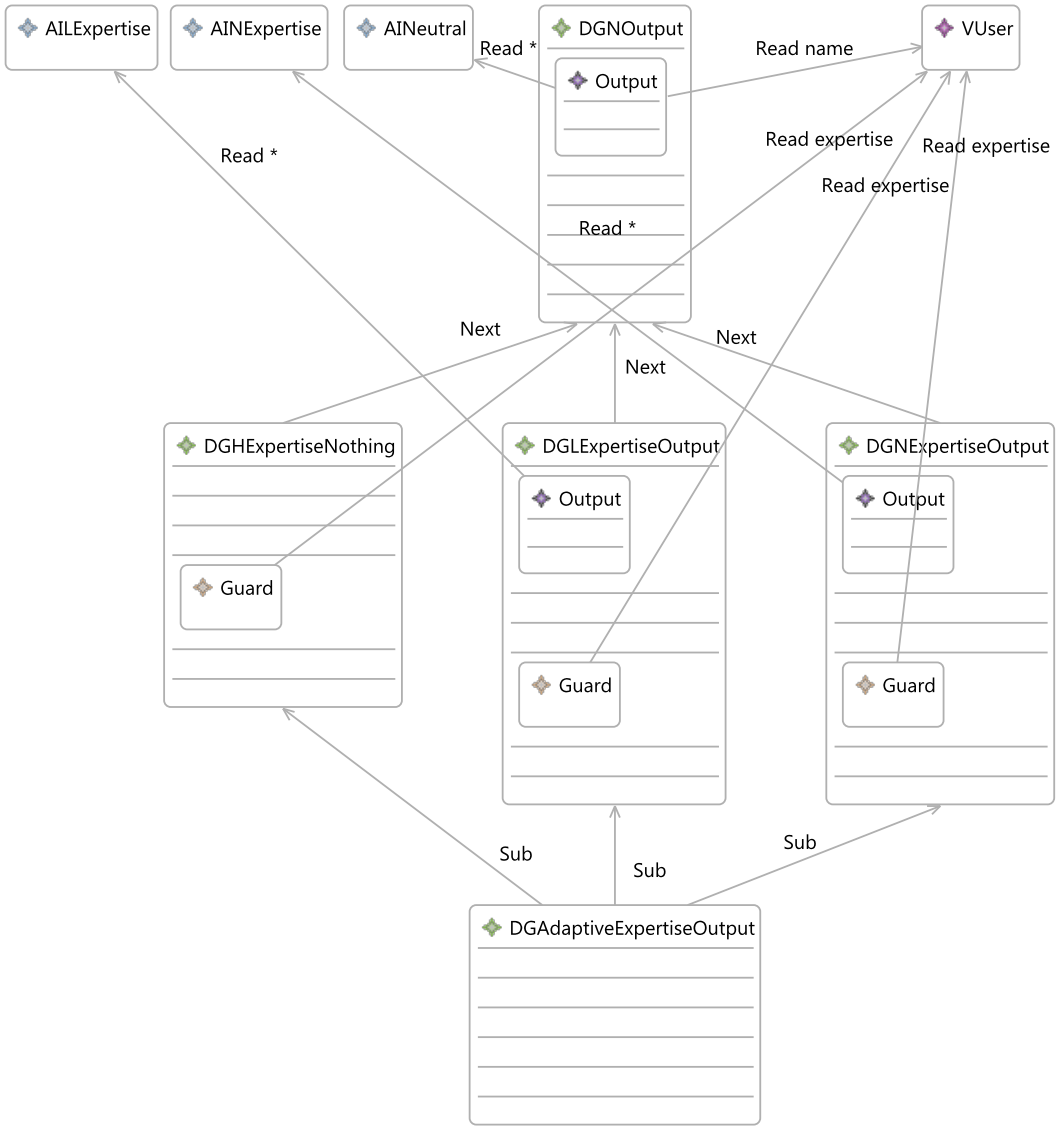

Figure 4.10: Expertise Adaptive Dialogue Part - A part of a dialogue that presents some output to the user, which is adaptive to his expertise level.

which reads the grammar for the user input from the abstract information *AIYesNo* and sets the value in the variable *VAnswer*. The next goal that is executed then is *DGDecision* which has two sub goals *DGYes* and *DGNo* both with a corresponding guard condition. The guard condition specifies that each of these goals can only come to execution if a suitable answer was given by the user. Then the dialogue will be continued with either *DGContinueYes* or *DGContinueNo* which will be replaced with dialogue goals from the parent dialogue structure where this dialogue widgets is included as a sub dialogue.

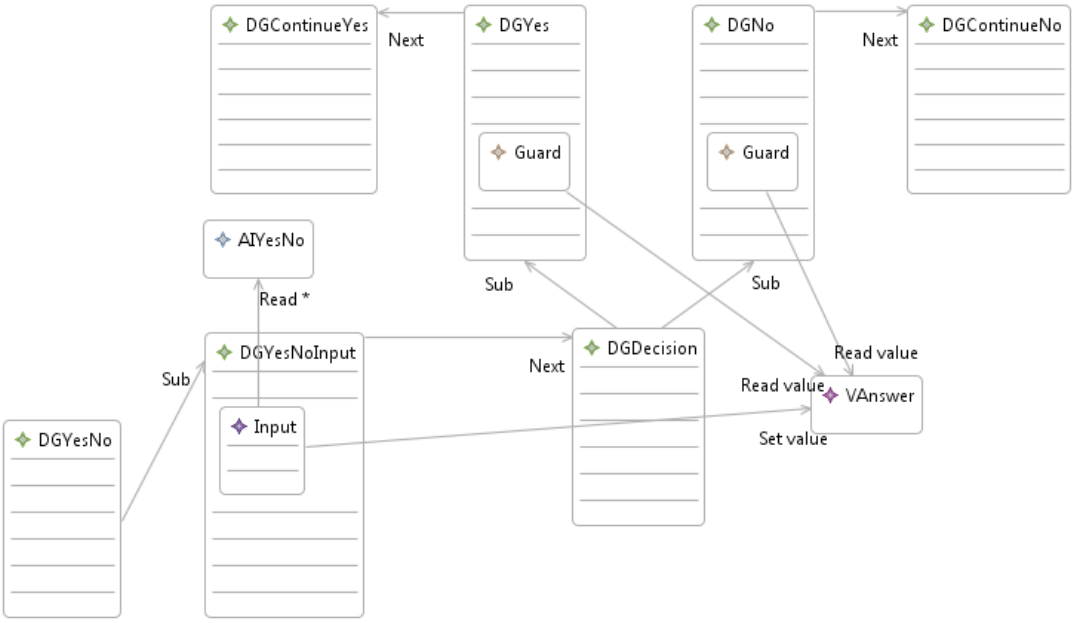

Figure 4.11: Dialogue Part for Yes-No Answer - A dialogue widget describing the input for a simple question which can be answered by yes or no. This widget is very widely useable for any kind of input that needs to be explicitly confirmed by the system.

4.1.3.1 Widget Accumulation

In figure 4.7 an example of a sub dialogue for output adaptive to the user's sensor estimated level of *Pleasure* is shown. *Pleasure*, *Arousal* and *Dominance* are dimensional measures for (user) emotion like in (Mehrabian, 1996).

There may be more variables that should influence adaptive output in the same way but may in combination need a special kind of output. That is why we propose a method

of merging different dialogue widgets into a single one combining the sensor variables that should guide the adaptive output. We may also want to adapt the dialogue output to *Arousal*, *Dominance* and the user's *Expertise*. By the term *Expertise* we understand the user's knowledge about how to use the system. For this purpose different sub files containing similar goal structures can be molten into one containing all the alternatives that arise from combinations of the guard conditions. Taking a look at figure 4.12, one can see how different sub files with the same structure can be merged taking into consideration the conditions specified in the guards. On the left side you can see two sub dialogues of which the top goals are named *Greet*. The upper *Greet* dialogue goal has the sub goals *Low Pleasure* and *High Pleasure* which can be accessed depending on the P value which may again stand for the user's estimated emotional situation in the pleasure domain. The lower dialogue goal *Greet* has also two sub goals *Low Arousal* and *High Arousal* which can be accessed according to the value of A which may stand for the user's arousal. If the upper and lower *Greet* dialogue goals are now combined as sub goals in the same parent goal it may be desirable that the conditions in the guards will be combined. This is shown on the right side of the figure. The sub goal *Low Pleasure* is combined with *Low Arousal* and *High Arousal* in the goals *Low Pleasure_Low Arousal* and *Low Pleasure_High Arousal* as sub goals of the dialogue goal *Greet*. In the same way *High Pleasure* is combined with *Low Arousal* and *High Arousal* in the goals *High Pleasure_Low Arousal* and *High Pleasure_High Arousal*. Each guard condition of the first goal is combined with a logical "&" with the guard condition of the second goal.

To make this abstract example a little more practical we can imagine that the dialogue model might react to low pleasure with trying to cheer the user up which would be specified in *Low Pleasure* while it might greet the user happily if his pleasure level is already high (*High Pleasure*). A tired user with a low arousal level might be motivated by the system to go on with the dialogue (*Low Arousal*) while an already agitated user might be calmed (*High Arousal*). Combining the combinations in the merge process we would need to specify an appropriate system utterance for a sad and tired user (*Low Pleasure_Low Arousal*), a unhappy and aroused user (*Low Pleasure_High Arousal*), a happy and tired user (High Pleasure_Low Arousal), and a happy and aroused user (High Pleasure_High Arousal).

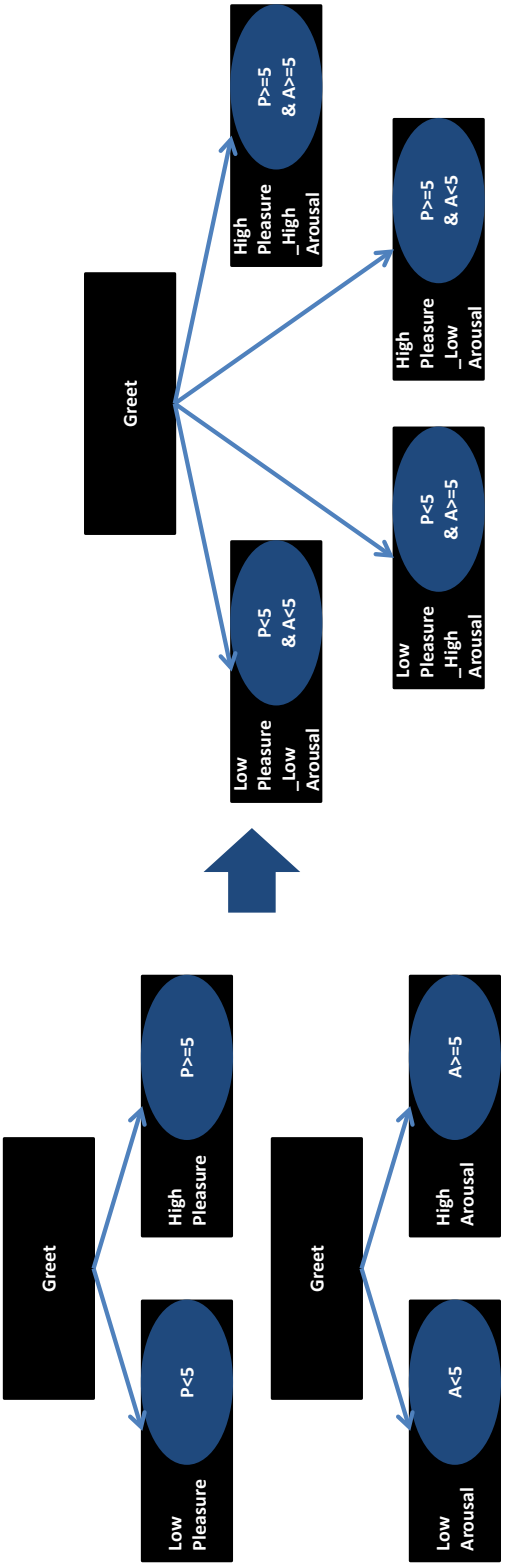

Figure 4.12: Two sub dialogues from separate files are merged together combining their guard conditions.

By this form of merging similar kinds of adaptivity can be easily combined but there are some difficulties to be overcome when using this approach. The sub goals that may be combined can e.g. have the function of presenting output to the user. Therfore a decision must be made on what happens to the output and how to merge it. For this question we found a simple solution by externalizing these kind of resources (like e.g. output), which will be presented in the subsection 4.1.3.2.

In different contexts it may be desirable to merge the goals in another way. Suppose we want to add dialogue goals with different kinds of functionality but adaptive in a similar way as above. Here it would not be desirable to have the goals merged in the manner just described above because of the difference in functionality. We would like to just add the additional functionality to the dialogue goal. In order to make this kind of merging more clear in figure 4.13 there is a short example how merging in this case can be done. Imagine that it would be inappropriate to create a special *GoodBye* statement for a user with a low pleasure level and a low arousal level. One would like to use the *GoodBye* statements used for an user with low pleasure and used for an user with low arousal sequentially. One would also like to do the same for the other combinations analogously. Then one may rely on the merge process described in figure 4.13.

However, widget accumulation can also mean that widgets are combined over several levels as we can see in figure 4.6. It references the dialogue widgets specified in figures 4.7, 4.8, 4.9, and 4.10 in the dialogue goal *DGQuestionOutputSG*. It also references another widget for the confirmation part of the dialogue which is shown in figure 4.14 in the dialogue goal *DGConfirmQuestionInputSG*. Here the dialogue goals *DGContinue* and *DGBack* are also contained in the sub file with the confirmation widget. They are replaced with the corresponding dialogue goals from the parent goal in which the sub file was included. The dialogue goals are then removed from the parent goal as they are now integrated into the structure of the sub dialogue. The widget from figure 4.6 could also be included in another widget. In this way it is possible to build very complex widgets without losing clearness.

4.1.3.2 Resource Externalization

In our dialogue model we externalize the resources that are used by the dialogue goals. One can see this in figure 4.7 where the abstract information objects *AIHP*, *AILP*,

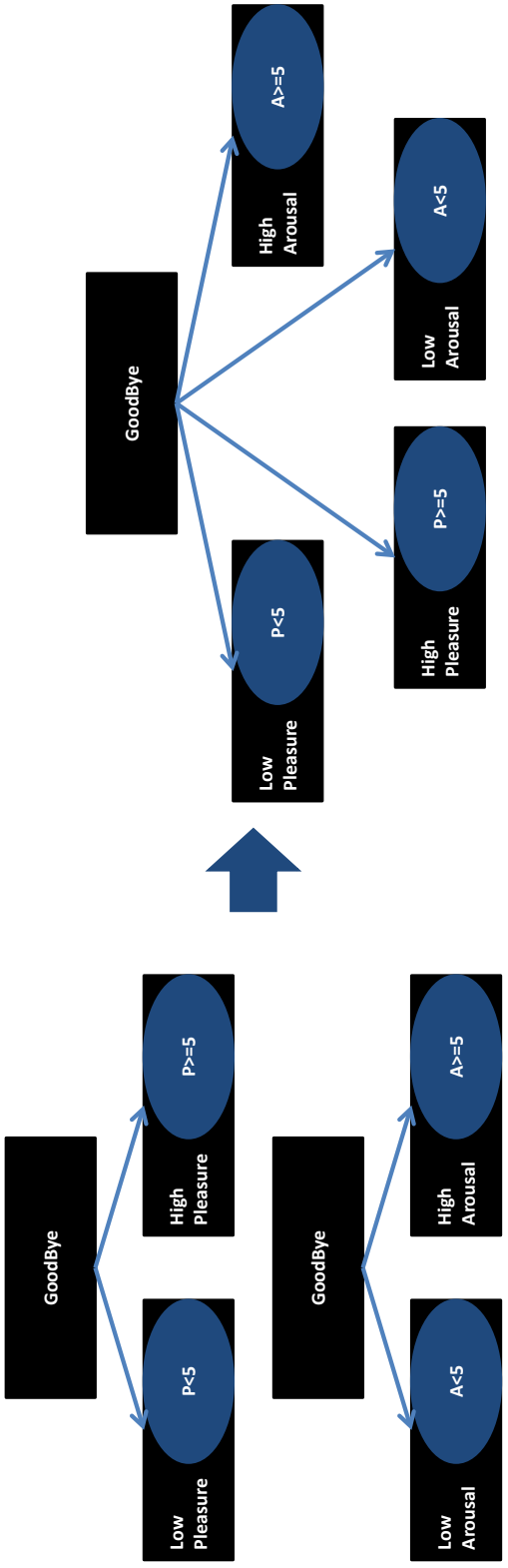

Figure 4.13: Two sub dialogues from separate files are merged together adding their subgoals without combining them.

Figure 4.14: This dialogue widget presents different forms of confirmation possibilities. Depending on confidence of the speech recognition and the user's expertise either explicit confirmation, implicit confirmation or no confirmation will be used as confirmation strategy. For explicit confirmation the dialogue widget shown in figure 4.11 is used.

and *AINeutral* reference resources that are specified in separate files. In figure 4.6 the resources used for mapping the user input to dialogue variables are externalized through the abstract information object *AIQuestion* and the variable *VQuestion*. By externalization of our resources, we achieve a higher degree of flexibility. We can easily exchange one resource for another. Furthermore, the merging of dialogue goals is possible because their resources don't have to be merged in a linguistic merging process (which might prove difficult). They are simply referenced by the dialogue goals that are the input for the merging process and are referenced by the dialogue goals that are the output of the merging process as well.

4.1.4 Summary for Dialogue Model

In this section we presented the structure and semantics of our dialogue model. Introducing the concept of guards and hierarchical goals we showed how semantic units may be grouped and adaptivity may be achieved. In addition, we showed how this model allows to let the user choose how to continue with the dialogue. We explained dialogue widgets which make the dialogue model reusable and portable and therefore reduce modeling costs. In this context we showed how to split the dialogue model over several files which is good for clarity. Novel and unique qualities of this dialogue model are its combination of goals and guards in a hierarchical structure. Based on this, multimodality, concepts like dialogue widgets, structuring the model into separate files, and above all adaptivity become possible.

4.2 Dialogue Control

Deciding at runtime about the way how the dialogue is continued dialogue control algorithms need to consider several parameters. In non-adaptive dialogue there may be the user's input, the dialogue history, and most of all dialogue variables that may have been initialized by the user or the application that he wants to access through the dialogue system. Adaptivity makes this decision process more complex because context parameters now influence the decision process as well. In addition to that, the structure of the dialogue has increased in complexity which may also increase the burden on the logic controlling the sequence of the dialogue steps. In this section we

address the aspects of dialogue decision and logic under the premise of adaptivity and modularity.

4.2.1 Initialization

Because of the modular concept of the dialogue manager, which means to let the dialogue designer specify in a graphical editor what may happen during the dialogue at runtime, it is possible to initialize the dialogue control with the file generated from the graphical editor. In this process java objects are created from xml and a dialogue goal tree is created in memory which is traversed at runtime. For this purpose we rely on XSL transformations to create the java classes from a template that is defined in the graphical editor. This template displays all possible connections and compositions of the domain model. That is why it is possible to infer all information needed in the java classes that will be instantiated with domain model objects in order to let the dialogue control work at runtime. The process from graphical editor to java objects is outlined in figure 4.15.

When the dialogue domain is instantiated in memory the dialogue may be started. The dialogue processing relies on two principles. On the one hand the hierarchical dialogue model is evaluated according to the different forms of dialogue goals which are user initiative, system initiative and proactive dialogue goals.

4.2.2 Hierarchical Dialogue Goal Processing

As there are different kinds of dialogue goals which may be used for specific purposes there must be a prioritization for execution at runtime. This prioritization completes the hierarchical organization of the dialogue model. For processing the hierarchy the dialogue control starts at the goal which is at the top of the dialogue tree in memory. From there it descends until it finds dialogue goals which contain either a guard condition or an input or output object. These goals are under discussion for execution at the moment. As there are three different kinds of goals there also may be three different kinds of goals under discussion.

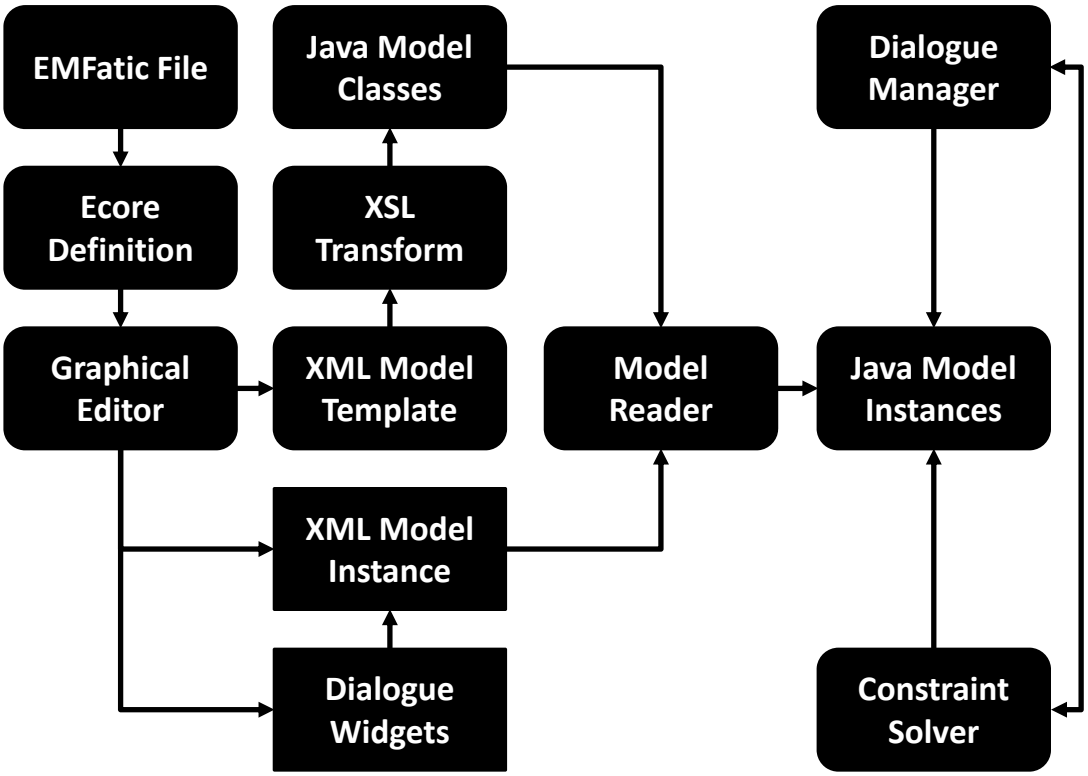

Figure 4.15: Outline of the process from model to runtime objects - The starting point for the graphical editor is the *EMFatic File* from which an *Ecore Definition* is generated. Based on this *Ecore Definition* which contains all information on the structure of the dialogue model a *Graphical Editor* may be produced as described in chapter 4. With this *Graphical Editor* an *XML Model Template* as well as the *XML Model Instance* and *Dialogue Widgets* are created. The *XML Model Template* contains an example describing the structure of the dialogue model in a form that may be easily turned into *Java Model Classes* by the use of an *XSL Transformation*. The *XML Model Instance* contains all neccessary domain information that are needed at runtime. This is complemented with reusable information, so called *Dialogue Widgets*. The purpose of the *Model Reader* is to create *Java Model Instances* from the *Java Model Classes* using the *XML Model Instance* Information. The *Java Model Instances* are then used by the *Dialogue Manager* and the *Constraint Solver* which make up the dialogue control.

User Initiative Goals

Goals which are tagged to be user initiative may contain an input object which activates them if the user uttered any matching phrase. They may also contain a guard condition that will need to be fulfilled in order for the dialogue goal to become executable. User initiative goals are only activated if the user utters a matching phrase. They will be kept under discussion if the dialogue control descends further in the dialogue tree, but they will be discarded from being under discussion if the dialogue control ascends higher than their containing parent goal.

Proactive Goals

Goals which may be presented proactively to the user are the second kind of goals which are considered for execution. As the system initiates them it doesn't make sense to put input objects into these kind of goals. If there is one or more proactive goal executable at the moment they will be executed in the order in which the dialogue control found them in the dialogue tree. Proactive goals which are not executed at the moment will be available and valid if the dialogue control descends further down in the dialogue goal tree. If the dialogue control however ascends higher in the hierarchy than their position they will be removed from being under discussion.

System Initiative Goals

System initiative goals are executed after all executable user initiative goals and proactive goals have been executed. Their executability depends on their guard conditions and upon their being under discussion meaning their position relative to the position where dialogue control is at the moment. From the position of dialogue control, the active dialogue goal, all system initiative goals that are subgoals and contain guard conditions may be selected and added for being under discussion. If the active dialogue goal has already decreased the maximum of executable subgoals to zero in the process of the dialogue, no subgoals will be selected for being under discussion any more. Instead, dialogue control will try to find a next goal that is executable or rise in the dialogue hierarchy until a goal is found containing other subgoals to be executed or having a next goal link which has not yet been visited. In this order first any subgoals which

may be under discussion will be executed according to their guard conditions and then the next goals will be considered by dialogue control.

4.2.3 Constraint Programming

For the evaluation of the different context conditions that influence the adaptive dialogue we state these conditions as constraints. By restricting the form of constraints to constraints on integer variables we may use a finite domain constraint solving approach. A description of finite domain constraint programming may be found in (Fernández *et al.*, 2006). The main idea is to find out possible configurations for variables which may take a finite amount of values by means of constraint logic programming (cf. Bartak (1999)).

In order to find out which of the dialogue goals that are under discussion at a certain time in the dialogue are executable their guard conditions are evaluated. The conditions in the guards are parsed and for each dialogue goal under discussion a set of constraints is created which contains a logical "AND-condition" of a variable representing the number of the dialogue goal with the condition that is contained in the guard. All dialogue goal "AND-conditions" are brought together using logical "OR-conditions". These "OR-conditions" are linked with an "AND-condition" to a set of "AND-conditions" representing the dialogue variables and sensor variables that are already initialized. The process of creating the constraints describing the dialogue state at the moment is shown in an example in figure 4.16. Using this constraint the constraint solver may directly infer which goals are possible according to the variable configurations. For constraint solving we use the choco constraint solver (cf. Jussien *et al.* (2008)).

4.2.4 Algorithm Description

Traversing the dialogue tree is done in two modes. Either the dialogue control descends and executes dialogue goals or the dialogue control ascends and looks for dialogue goals which haven't yet been executed. While going down the dialogue manager continues its descent until it finds a dialogue goal which has a maximum number of subgoals to be executed that is less than one. If there is a next goal which has not yet been executed it will follow this relation and continue its descent in the same way. Then it switches to going up until it finds a goal which has a maximum number of subgoals to

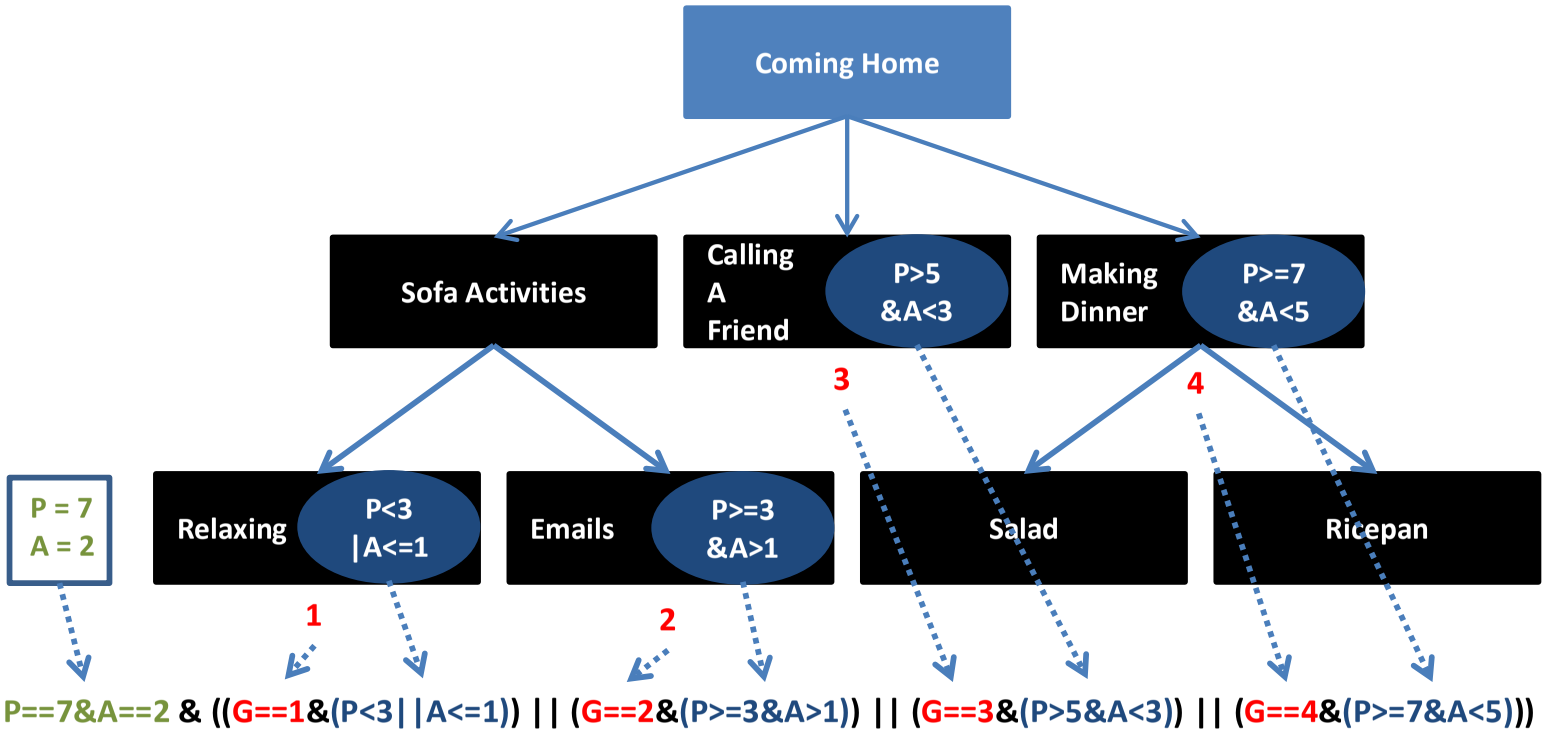

Figure 4.16: When the dialogue manager needs to decide upon the guard conditions it transforms the variables which are initialized and the dialogue goals which are under discussion (*Relaxing*, *Emails*, *Calling A Friend*, and *Making Dinner*) into constraints. The initialized variables displayed in green color are transformed into equality constraints. The goals which are under discussion are numbered (red numbers) and an auxiliary variable G is introduced. The variable G is required to be equal to the numbers of the goals. These equality constraints are brought together with the respective guard conditions using "AND-conditions". The guard conditions are directly translated into constraints (blue color). In this way there is an "AND-condition" for each goal. They are put together using "OR-conditions". Together with the variable conditions the constraint solver will eliminate the conditions which may not be fulfilled and return the possible variable configurations to fulfill the constraint. In this way the possible configuration of the variable G will indicate which goals may be executed.

be executed that is greater than zero decreasing the numbers while going up from the executed dialogue goals. If it cannot go up it will try to go back next goal relations and mark the goals as already executed (setting their maximum subgoal execution value to minus one).

For traversal of the dialogue tree there have to be considered different kinds of dialogue goals which are saved in respective lists. These lists represent which dialogue goals are under discussion for execution at a certain point of time during the dialogue. As already explained the order of processing of the lists is that firstly user initiative goals are executed, then proactive goals are executed and finally system initiative goals. In each of these lists the guard conditions are evaluated. The executable dialogue goals are then saved into seperate lists which are freshly initialized for each dialogue step.

If there are more than one system initiative dialogue steps executable they will be offered to the user for choosing what he wants to do next. User initiative and proactive dialogue goals are executed in the order that they have been encountered by dialogue control in the dialogue tree.

For understanding the traversal of the dialogue tree in the following subsections we describe the algorithm in a more detailled manner. Before every dialogue step the dialogue control tries to go down, if this is not possible it tries to go up, and if this is not possible either it looks for system initiative dialogue goals which were saved because of the execution of user initiative or proactive branches having priority over system initiative goals.

The main dialogue loop is displayed in algorithm 1. This loop runs while no dialogue interrupt has occured which means that there was no user input activating an user-initiative dialogue goal and while the dialogue was not finished yet. The loop starts by selecting any special goals that need to be executed first in the order of user initiative goals (1.3), what next goals letting the user specify how he wants to continue the dialogue (1.5), and proactive goals (1.7). If any of these types of goals is pending, it will be selected as a current goal and the dialogue will continue at this position. If no goals are pending the dialogue tree will be traversed to a position where there are system initiative goals to be executed (1.9). From the selected current goal as a reference point all goals which should be under discussion are initialized (1.11). The sensor values are transformed into constraints (1.12) and then the executable dialogue goals are determined and executed (1.13).

Algorithm 1 dialogueLoop

```
1: while no Interrupt and dialogue not finished do
2:     if selectedUserInitiativeGoal ≠ ∅ then
3:         currentGoal ← selectedUserInitiativeGoal
4:     else if activeWhatNextGoal ≠ ∅ then
5:         currentGoal ← activeWhatNextGoal
6:     else if proactiveGoal ≠ ∅ then
7:         currentGoal ← proactiveGoal
8:     else
9:         moveOn
10:    end if
11:    initGUD
12:    initSensors
13:    executeDialogueGoals
14: end while
```

Algorithm 2 moveOn

```
1: if goDown is possible then
2:     goDown
3: else if goUp is possible then
4:     goUp
5: else
6:     endDialogue
7: end if
```

The dialogue control movement is shown in algorithm 2. It first tries to go down and follows next goal relations (2.2), then it tries to go up and follows next goal relations in a backward direction (2.4). Finally, if both are not possible the dialogue consequently has to be finished (2.6).

Going Down

Algorithm 3 goDown

```
1: function GODOWN    ▷ returns boolean value indicating if going down is possible
2:     if currentGoal still has subgoals which may be executed then
3:         return true
4:     end if
5:     if currentGoal has next goal which has not been executed then
6:         currentGoal ← nextGoal
7:         return true
8:     end if
9:     return false
10: end function
```

For going down in the dialogue tree the necessary steps are shown in algorithm 3. If going down is possible true will be returned to indicate this fact (3.3). If going down is not possible but there are next goals which may be executed the next goal will become the current goal (3.6) and true will be returned to indicate that moving was possible (3.7). If none of both movement directions were possible false will be returned in order to indicate this fact (3.9).

Going Up

In order to go up in the dialogue tree the required steps are shown in algorithm 4. As long as there is a parent goal or a previous goal (4.2) dialogue control will move to the parent goal (4.4) or to the previous goal (4.6). If there are sub goals true will be returned 4.9; if there are next goals dialogue control will move to the next goal and also return true 4.12. However, if dialogue control arrived at the top of the tree, false is returned in order to indicate that the dialogue has finished.

Algorithm 4 goUp

```
1: function GOUP                ▷ returns boolean value indicating if going up is possible
2:     while parentGoal ≠ ∅ or previousGoal ≠ ∅ do
3:         if parentGoal ≠ ∅ then
4:             currentGoal ← parentGoal
5:         elsepreviousGoal ≠ ∅
6:             currentGoal ← previousGoal
7:         end if
8:         if currentGoal has still subgoals to execute then
9:             return true
10:        else if currentGoal has a nextGoal to execute then
11:            currentGoal ← nextGoal
12:            return true
13:        else if currentGoal = topGoal then
14:            return false
15:        end if
16:    end while
17:    return true
18: end function
```

Collecting Goals Under Discussion

Algorithm 5 initGUD

```
1: function INITGUD
2:     if currentGoal has subDialogueGoals then
3:         for dialogueGoal ∈ subDialogueGoals do
4:             if dialogueGoal is a userInitiative dialogue goal then
5:                 userInitiativeGUD ← userInitiativeGUD ∪ dialogueGoal
6:             else if dialogueGoal is a proactive dialogue goal then
7:                 proactiveGUD ← proactiveGUD ∪ dialogueGoal
8:             else
9:                 initSystemInitiativeGUD(dialogueGoal)
10:            end if
11:        end for
12:    else
13:        initSystemInitiativeGUD(currentGoal)
14:    end if
15: end function
```

While traversing the dialogue tree, goals which may be considered for execution are added to GUD (goals under discussion) lists. This process is shown in algorithm 5. User initiative goals, proactive goals and system initiative goals are managed in separate lists because their priority for execution is different (5.5,5.7,5.9, 5.13).

Algorithm 6 adds recursively (6.6) all system initiative goals which have not yet been executed to the list of system initiative goals which are under discussion (6.5).

In order to evaluate the context information which is modeled in sensor values (e.g. sensors for emotion) these values are created as constraints which require the equality of dialogue variables to the measured values (7.4). These constraints are connected through logical "and"s and added to all runTimeConstraints (7.5).

Executing Dialogue Goals

Algorithm 8 shows how the dialogue goals are executed in their defined order. Before excuting each kind of dialogue goal their guard conditions are evaluated and the respective dialogue goals are put into lists for user initiative, proactive and system initiative goals which are executable (8.2, 8.3 and 8.7). Whereas user initiative goals are passive

Algorithm 6 initSystemInitiativeGUD

```
1: function INITSYSTEMINITIATIVEGUD(dialogueGoal)        ▷ initializes system initiative dialogue goals under discussion
2:     if dialogueGoal has not yet been executed then
3:         systemInitiativeGUD ← systemInitiativeGUD ∪ dialogueGoal
4:     else
5:         for dialogueGoal ∈ subDialogueGoals do
6:             INITSYSTEMINITIATIVEGUD(dialogueGoal)
7:         end for
8:     end if
9: end function
```

Algorithm 7 initSensors

```
1: function INITSENSORS                                  ▷ initializes sensor values as constraints
2:     for variable ∈ dialogueVariables do
3:         for field ∈ variable.fields do
4:             constraint ← ⟨field = sensorValue⟩
5:             runTimeCosntraints ← runTimeConstraints ∪ constraint
6:         end for
7:     end for
8: end function
```

Algorithm 8 executeDialogueGoals

```
1: function EXECUTEDIALOGUEGOALS ▷ executes dialogue goals according to their
   constraints
2:     execUserInitiativeGUD ← INITEXECGOALS(userInitiativeGUD)
3:     execProactiveGUD ← INITEXECGOALS(proactiveGUD)
4:     if execProactiveGUD ≠ ∅ then
5:         PROCESSEXECPROACTIVEGUD
6:     end if
7:     execSystemInitiativeGUD ← INITEXECGOALS(systemInitiativeGUD)
8:     if execSystemInitiativeGUD ≠ ∅ then
9:         PROCESSEXECSYSTEMINITIATIVEGUD
10:    else                    ▷ if dialogue domain designer made mistakes in modelling
11:        for dialogueGoal ∈ systemInitiativeGUD do
12:            set has been executed flag to true
13:            set maximum subgoal execution of parent goal to zero
14:        end for
15:    end if
16: end function
```

until activation, proactive and system initiave goals are directly processed (8.5 and 8.9). In case the dialogue designer made any mistakes specifying the number of sub goals which should be executed for the current parent goal this dialogue goal will not be visited again in order to avoid execution deadlocks (8.13).

Proactive dialogue goals are processed in the manner that is presented in algorithm 9. As they break the regular flow of the system initiative dialogue goals these goals are saved for continuing later (9.4). Then the proactive branch will be processed (9.5) and a flag indicating its recent execution will be set (9.6). In order to reactivate this proactive dialogue goal after a certain amount of time a thread may be started in order to unset this flag (9.7).

System initiative goals are processed slightly different from proactive dialogue goals as may be seen in algorithm 10. Dialogue control begins with checking if there are more than one dialogue goal executable at the moment (10.2). If yes the user is asked how to continue with the dialogue (10.3) and if no the single available executable system initiative goal is selected as the current goal (10.5). Before processing any input or

Algorithm 9 processExecProactiveGUD

```
1: function processExecProactiveGUD ▷ processes all executeable proactive goals under discussion
2:     for dialogueGoal ∈ proactiveGUD do
3:         if has not been executed then
4:             save system initiative dialogue state on stack
5:             execute this proactive branch of dialogue
6:             set has been executed flag to true
7:             start a thread to reactive proactive goal after specified amount of time
8:         end if
9:     end for
10: end function
```

Algorithm 10 processExecSystemInitiativeGUD

```
1: function processExecSystemInitiativeGUD ▷ processes all executeable system initiative goals under discussion
2:     if ||execSystemInitiativeGUD|| > 1 then
3:         ask user what goal he wants next
4:     else
5:         currentGoal ← execSystemInitiativeGUD.getGoal
6:     end if
7:     execute dialogue logic calls on enter
8:     if currentGoal has output or input then
9:         process any output or input
10:    end if
11:    execute dialogue logic calls on exit
12:    save currentGoal to dialogue history
13: end function
```

output for the current goal (10.9) its dialogue logic calls which should be performed when entering the goal are executed (10.7). Previous to saving the executed dialogue goal to the dialogue history (10.12) its dialogue logic calls which should be performed on exiting the dialogue goal are executed (10.11).

Evaluating Dialogue Constraints

Algorithm 11 initExecGoals

```
1: function INITEXECGOALS(goalsUD)        ▷ evaluates goal constraints and returns
   executable dialogue goals
2:     goalsUDIV ← new integer variable with codomain from 0 to ||goalsUD|| − 1
3:     goalUDI := 0
4:     orConstraints := ∅
5:     for goalUD ∈ goalsUD do
6:         orConstraints ← orConstraints ∨ {⟨goalsUDIV = goalUDI⟩ ∧
   goalUD.constraints}
7:         goalUDI ← goalUDI + 1
8:     end for
9:     constraintsToEvaluate ← runtimeConstraints ∧ orConstraints
10:    solve constraintsToEvaluate
11:    execGoals := ∅
12:    for all values for goalsUDIV which are solutions do
13:        execGoals ← execGoals ∪ goalsUD[goalsUDIV]
14:    end for
15: end function
```

For understanding the mechanism used to evaluate the currently executable dialogue goals with respect to the current variable configuration, the way in which the constraints in the dialogue goals are evaluated is shown in algorithm 11. First an integer variable is created which represents the dialogue goals under discussion by assigning them a number (11.2), then their guard conditions are connected by a logical “and” condition with the respective goal number (11.6). These conditions are connected by logical “or” conditions and finally the runtime constraints containing the current variable configurations are connected by a logical “and” condition to all dialogue goal constraints (11.9). After solving the constraint conditions for possible dialogue goals to execute

(11.10) these values are written into the list of executable dialogue goals (11.13). This list is then interpreted for continuing the dialogue.

4.2.5 Summary for Dialogue Control

In this section we described the mechanisms of dialogue control which focus on two principles: the hierarchical dialogue model and the constraints modeled in the guard conditions. After describing the initialization process of the dialogue control components we showed how the hierarchical dialogue model and the constraints are interpreted at runtime in a general way and also described the important parts of the algorithm which selects the next dialogue goal at runtime.

The algorithm not only allows to select adaptive dialogue goals based on context information, but also allows the user to select the next dialogue goal if more than one is possible. Another advantage of this abstract handling of context information is that any possible context information which may be gathered by sensors if reduced to the guard condition form may be used for adaptive dialogue.

4.3 Dialogue Design

To avoid any confusion of different terms in this chapter we use the term dialogue design in order to refer to all the aspects of designing a domain, including the practical realization thereof. The term domain design refers to creating an instance of the dialogue model which describes the particular dialogue goals that are provided for realizing a dialogue application.

Regarding dialogue design there are different aspects to be considered in the context of adaptivity. One major aspect is the increased size of the domain and the needed effort to create the model. Therefore, it seems to be important to think about possibilities how to make it easier for the designer to create the domain.

4.3.1 Domain Design

The main influences on domain design are the form and the features of the application that the dialogue system will provide access to. For example in an automotive context, where the dialogue system should only be used to control the radio and some car controls like heating a simple dialogue domain will be enough. However, if the application is an

intelligent calendar that may be used to organize appointments and activities allowing to rearrange appointments while satisfying causal conditions among them it is quite different. There will be reasoning needed to avoid timing conflicts and to propose appointment schedules. As already mentioned above there may also be even more complex domains if the dialogue system is based on a multi-agent architecture. In many cases the dialogue system development is application driven which means that its focus lies more on how to make the dialogue system capable of application related features than on providing a domain design concept that is reusable and intuitive for the designer. That means that the domain designer not only has to learn about the internals of the dialogue system but also is often limited to a certain kind of application. Considering domain design there is still potential to improve on generality and on intuitivity.

4.3.2 Adaptive Domain Design

Out of the desire for adaptive dialogue grows the need for larger domains which should allow for different variants of the dialogue. Looking at these requirements for dialogue design the question arises how to alleviate the burden that lies on the dialogue domain designer. In many cases the domains that have to be modeled are difficult to understand and require the dialogue designer to know about the internal workings of the dialogue control. We may find agent architectures which are fairly complex in their functioning and also in the way that the dialogue should be modeled. An important example for agent based architectures is the TRAINS/TRIPS system (see (Allen *et al.*, 2001)). Another approach is the information state approach of trindikit by Larsson & Traum (2000). In trindikit a prolog-like dialogue domain design was used. Another even more elaborate approach is to use partially observable markov decision processes as a dialogue design basis like in (Bui *et al.*, 2009). This kind of modelling normally requires large corpora of speech data to derive a dialogue domain design that works.

While these systems allow complex dialogues and application domains their approach to modeling the domain is not optimal for the domain designer. The dialogue design is not clearly separated from the dialogue control which means that the dialogue designer needs to learn about how the dialogue control works in order to create a dialogue domain.

For example trindikit relies on a specific prolog syntax that the designer has to learn in order to create a dialogue domain.

In the face of adaptive dialogue domains which are more complex and larger it would be best to provide the designer with an appropriate dialogue model. If it is sufficient to know the semantics of the dialogue model. The designer can create the domain without deep knowledge about the dialogue control mechanisms. The work process can then be split. The designer may prepare the dialogue domain while the dialogue programmer may take care of dialogue control. In order to provide this opportunity in a way that is comfortable for the dialogue designer it is neccessary to specify a common language as a dialogue model that may on the one hand be understood by the dialogue designer and on the other hand be interpreted by the dialogue control. For the dialogue designer a comfortable way to specify the dialogue model might be to use a graphical editor to create the dialogue from predefined construction elements.

In order to realize an editor which provides the dialogue designer with this opportunity and at the same time let's the dialogue control work with the output of the editor it is advisable to apply a commonly used format. This will allow the exchange of either dialogue control or the editor in order to be flexible to changing requirements.

4.3.3 Graphical Dialogue Domain Editor

For our generic approach to dialogue modeling we recommend XML as a data format in order to be flexible during editing and processing. The usual approach when aiming to define an XML-based data format would come along with the use of an XML Schema, a RELAX NG description or a DTD. It is best practice to utilize such a schema definition language in order to restrict the wide expressiveness of XML to a smaller subset that fits the needs of modeling a dialogue (cf. Becher (2009); Van Der Vlist (2004)).

The shortcomings of this approach are that XML Schema as well as RELAX NG are quite complex. While DTD has a somewhat simpler syntax, it is not XML itself and its expressiveness is limited when it comes to data types and power of semantics.

With a given schema it is possible to validate the XML files. One can verify if the XML conforms to the restrictions made in the schema. Therefore, to fabricate valid XML, the dialogue designer still needs to know the syntax as well as its semantic meaning, partly encoded in the schema, partly ascertained by common sense. So this is no trivial business at all.

For usability reasons it is desirable that a dialogue designer is given a simple interface, which provides the opportunity to model a dialogue that can later be interpreted by the dialogue manager – all without the the need to understand the underlying schema. In addition the interface shall also provide information about what is allowed as valid syntax for the dialogue model and what is not.

As XML editing can be tiresome and is often not as intuitive as using a graphical editing framework, we wanted to have an application which enables the designer to work and model the dialogue in a graphical way.

As a result of our search for a graphical editing tool we propose the use of the eclipse graphical modeling framework[1]. This proposal comes with the idea to use the eclipse modeling framework as a basis to generate a graphical editor. We will now describe the process in detail.

The eclipse modeling framework (EMF) provides an opportunity to model a class structure using a subset of the UML2 standard (cf. Steinberg *et al.* (2008)). If necessary, a designer can make use of a diagram editor which allows to oversee the class structure. But one can also use a plugin called *emfatic*[2] which provides a very simple and easy description language for these classes to be defined. The differences in the description can be seen when comparing figure 4.17 and figure 4.18, keeping in mind that the ecore file is edited graphically and that the *emfatic* file is a very concise textual notation.

Based on this description language it is possible to create all the prerequisites which are needed for a graphical editor that allows for creating instances of the classes defined in EMF. The resulting graphical editor provides the opportunity to arrange instances of classes defined in the class structure in exactly the same way as their relationship was defined in EMF. An editor created this way restrains a dialogue designer from invalid modeling by simply just allowing valid modeling operations.

The next step is to generate an XML description file incorporating the class structure based on the UML diagrams modelled earlier. From this so called ecore file you can derive a diagram file that allows you to review the class structure graphically. Figure 4.19 depicts an example of such a dialogue class structure.

If satisfied, one can generate an eclipse plugin that allows to create instances of the classes of the self defined class model in a graphical front end.

[1] `http://www.eclipse.org/modeling/gmp/`

[2] `http://www.eclipse.org/modeling/emft/?project=emfatic`

```
1 @gmf(fu="bar")
2 @namespace(uri="dialoguemodel", prefix="dialoguemodel")
3 package dialoguemodel;
4
5 @gmf.diagram(fu="bar")
6 class DialogueModel {
7   val DialogueGoal[*] topGoal;
8   val GoalLink[*] goalLinks;
9   val VariableLink[*] variableLinks;
10  val Variable[*] variables;
11 }
12
13 @gmf.node(label="name")
14 class DialogueGoal {
15  attr String name;
16  attr String description;
17
18  @gmf.compartment(fu="bar")
19  val Utterance utterance;
20
21  @gmf.compartment(fu="bar")
22  val Grammar grammar;
23
24  @gmf.compartment(fu="bar")
25  val Effect effect;
26
27  @gmf.compartment(fu="bar")
28  val Guard guard;
29
30  @gmf.compartment(fu="bar")
31  val ApplicationLogic[*] applicationLogic;
32 }
```

Figure 4.17: Example of EMFATIC language - Structure of the dialogue model written in the *emfatic* language. The model is based on dialogue goals which impose a certain hierarchy and flow on the dialogue. These dialogue goals can contain utterances or grammars indicating if a certain goal is used for input or output. Guards and effects act like temporal constraints and define the arrangement of the different dialogue goals within a dialogue flow.

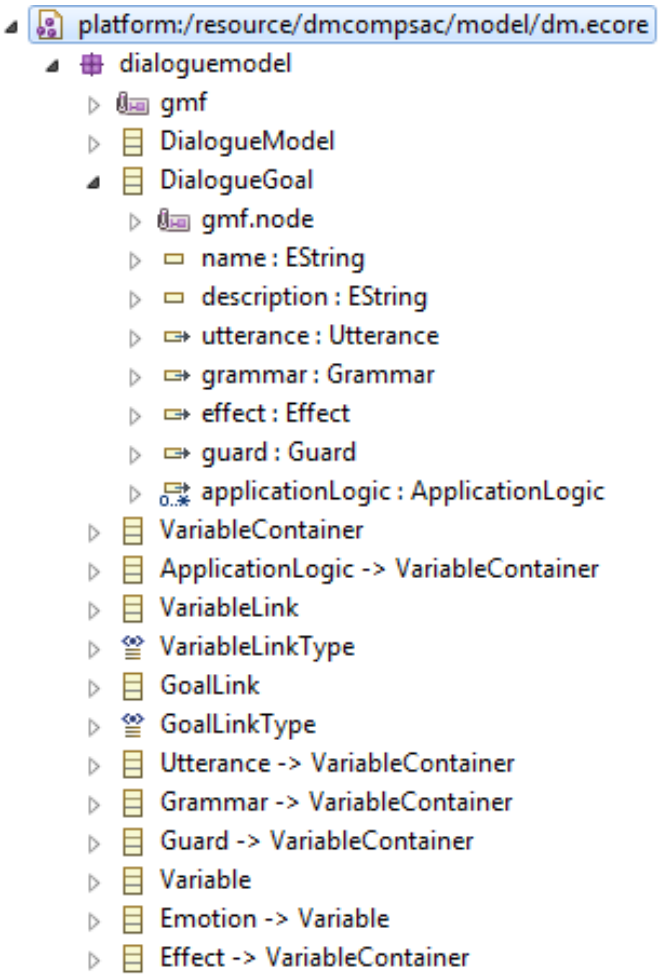

Figure 4.18: Example of ECORE editor - Part of the dialogue model forming an exemplary basis for the generation of a graphical dialogue model editor described in the ecore format.

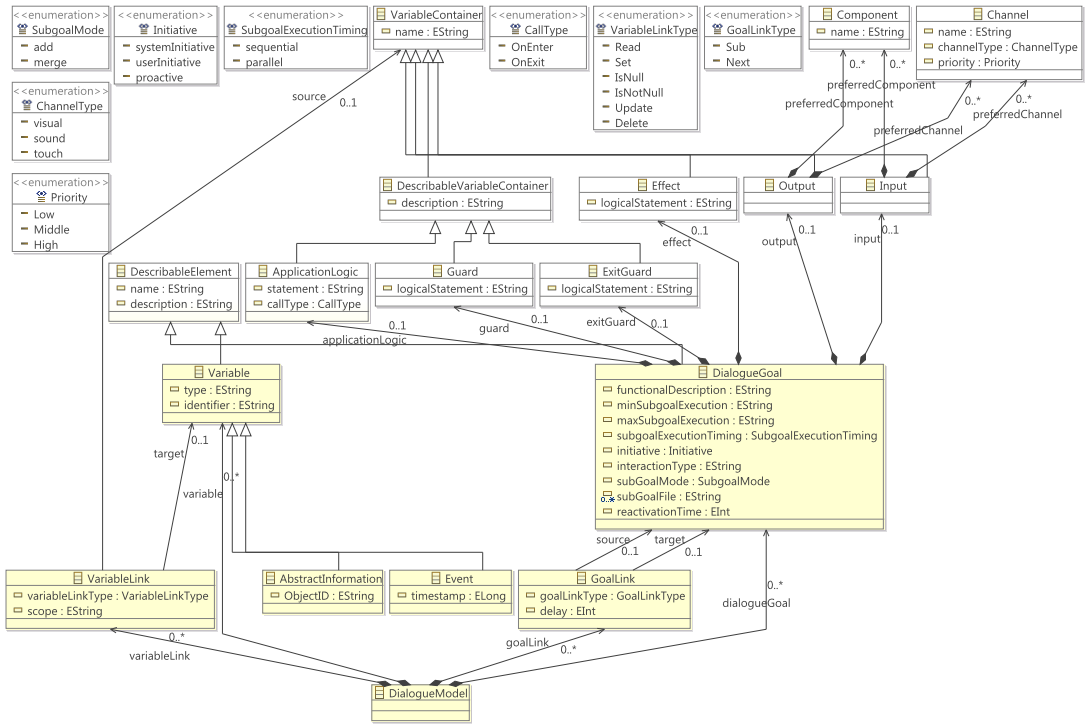

Figure 4.19: Graphical Editor - An example of the graphical editor for the class structure in EMF. The diagram depicts the structure that defines the rules for the creation of the dialogue model in terms of the UML2 syntax.

The idea put in other words is to use EMF to define the structure of the dialogue model in order to have a graphical editor for the dialogue designer. The graphical editor is easy and intuitive to use and allows the dialogue designer to create a domain model without knowing about the internal matters of dialogue management. At the same time the domain model is restricted to a structure that is guaranteed to be interpretable by the dialogue manager.

In this way there is a clear separation of model and algorithm not only in the software design but also in the software design process. This allows to distribute the tasks for developing the dialogue system. There can be a dialogue model design team that is primarily concerned with the dialogue flow and there can be a dialogue management team that is concerned with the communication between dialogue manager and applications as well as the interpretation of the dialogue model.

4.3.4 Proof of Concept

In order to provide a demonstration of the described method for dialogue modeling we outline the steps by an example for a multimodal dialogue model based on Nothdurft *et al.* (2010).

The first step is to define this desired structure in an *emfatic* file. An example is illustrated in figure 4.17 also revealing the conceptual design of the exemplary dialogue model. For this *emfatic* file to be translated the *emfatic* plugin is needed.

Theoretically it would also be possible to directly create an ecore file (as can be seen in figure 4.18) in which you specify the structure of your classes accordingly but as stated above, *emfatic* has a more compact syntax. From the *emfatic* file you generate an ecore model describing the specified structure in an XML file format. With the help of this ecore XML file you can generate a diagram editor that lets you review your defined class structure in a more convenient and graphical way. An example diagram is shown in figure 4.19.

The structure of the dialog model has the form of a class diagram and specifies the relations between the different classes. These classes form the basis for an instance of a dialogue model which is then composed of instances of the classes specified here.

If one is satisfied with the setup of one's class diagram it is now possible to make use of another eclipse plugin called *eugenia*[1] to generate all the necessary files and

[1] `http://www.eclipse.org/gmt/epsilon/doc/eugenia/`

classes that are needed for a graphical editing tool of the defined class structure. The *eugenia* plugin is not necessary but it helps a lot in simplifying the complex process to generate a graphical editor from the class diagram which represents our dialogue model. The complexity of the generation process makes it possible to customize the graphical editor to one's own needs in many ways. But for our purposes it is sufficient to use the *eugenia* plugin without any special customizations. The editing tool that comes out in the end can be exported as an eclipse plugin and deployed to the dialogue designer.

We now briefly describe the generation process of the editing tool itself. First you generate a domain generator model from the ecore file. Using this model you can then generate a set of java classes that form the basis for the editor that we want to obtain. Then you generate a graphical definition model, a tooling definition model and a mapping model that combines both of the definition models. Using these, you generate a diagram editor generator model that needs to be resynchronized to your data. From this model it is then possible to generate the java code for the diagram plugin using the generated files. You can see a list of these model data files in figure 4.20.

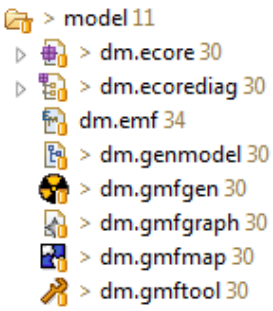

Figure 4.20: Files for generating Graphical Editor - Model data files required for the generation of the graphical editor. While the whole model is specified and contained in the .ecore or .emf file the .ecorediag file helps to understand the structure of the model. Specifying the steps of the generation process, the .genmodel, .gmfgen, .gmfgraph, .gmfmap, and .gmftool files may be customly edited in order to fine tune the behaviour of the generated graphical editor.

From the generated java classes a deployable plugin can be exported. With this plugin it is possible to create instances of the classes defined in the ecore file in a graphical editor. The emerging diagram satisfies all constraints that are formulated in

the ecore file. An example diagram file can be seen in figure 4.21. As defined, there are different connection types in the palette to model relations between different goals or goals and data.

At the end of the process the diagram file has an underlying data file which represents the dialogue model in XML. While the diagram file itself specifies the arrangement of the diagram for the model and is also in XML format, it is not needed for interpreting the dialogue model.

4.3.5 Summary of Dialogue Design

In this section we discussed a practical method to decouple dialogue design from the internal logic of the dialogue manager. Following our advise, a dialogue designer can operate freely using a quickly and easily generated graphical editor, customized for his individual needs. This custom-tailored editor allows for the valid specification of dialogue flow. Learning to understand the internals of the dialogue management algorithms is hardly necessary any more. Because the graphical editor constrains the dialogue designer to build a valid dialogue model, the model is guaranteed to be interpretable by the dialogue management at runtime.

4.4 Conclusion

In this chapter we presented our tripartite dialogue management approach with dialogue design, dialogue model and dialogue control. While the designer may use the model without knowing about the internal logic of dialogue control, dialogue control may build on the standardized form of the dialogue domain to bring the dialogue to execution at runtime. We presented the graphical editor which allows the designer to easily create adaptive dialogue domains. Created with the editor the dialogue domain automatically has the form specified in the model and can be read by dialogue control. We also showed how dialogue control traverses the tree-like dialogue model in order to create adaptivity at runtime.

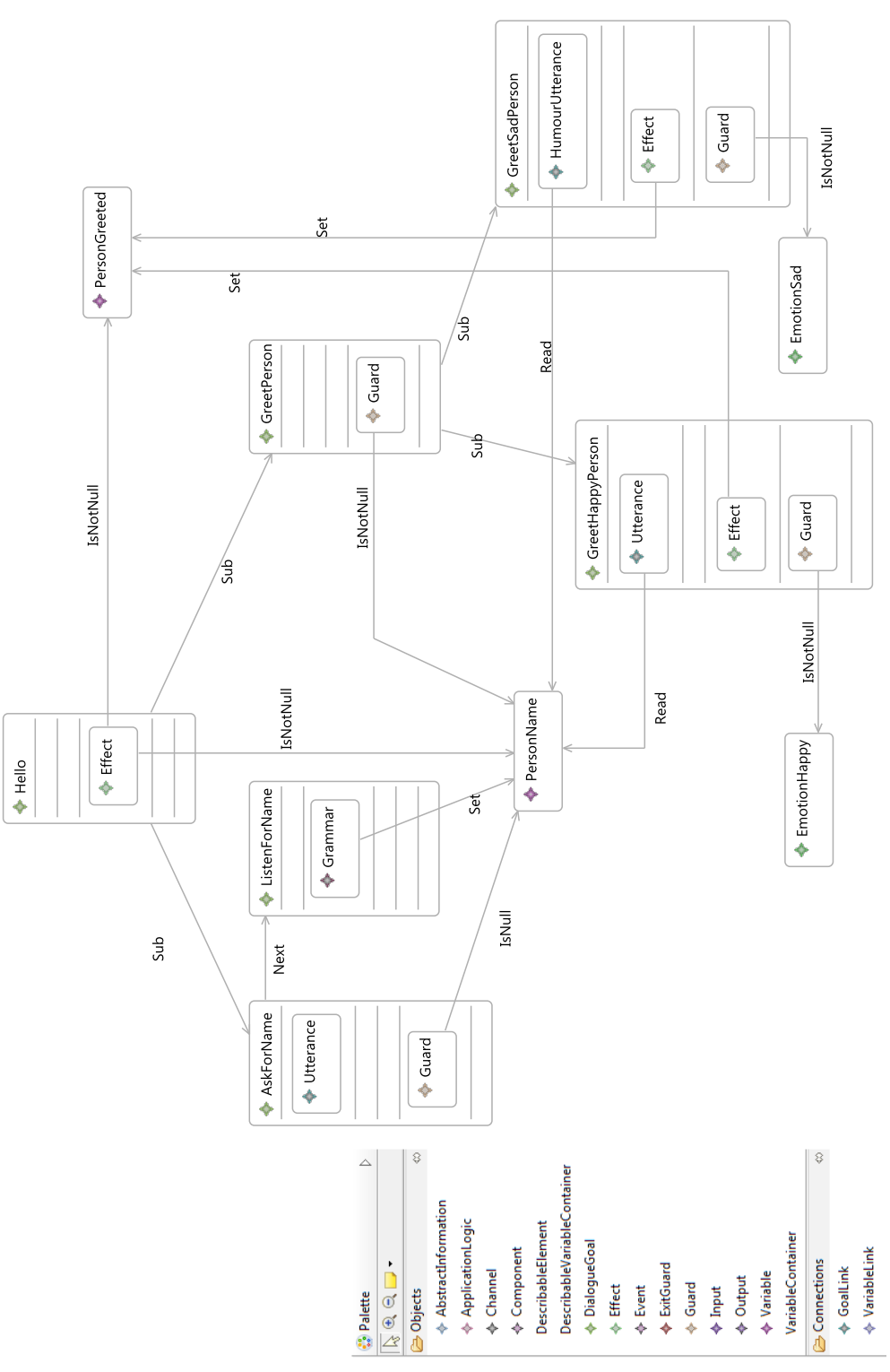

Figure 4.21: An example of the graphical editor resulting from the definition files. The dialogue model can be edited, satisfying the underlying structure defined in EMF. On the left *Palette* you see the parts of the dialogue model that you can create instances of in the editor. In the illustrated scenario you can create links (*Connections*) that mirror the relations between the instances (*Objects*) and arrange the whole structure on the canvas.

5

Evaluation

In order to evaluate the adaptive dialogue management approach different quality criteria have to be considered. One aspect is the portability meaning how easily components may be reused to create different dialogue system applications and how easily the components may be used for different modalities. This aspect was evaluated by designing two different applications of which one was a multimodal application which should help to create the correct cable connections for a home cinema system. This proof of concept showed that the dialogue management architecture proposed was useful and working. As a further proof of concept we implemented another application. The other application was a spoken dialogue system application which served as a scenario for prospectively evaluating the different adaptive features with user tests. This scenario was to book a flight from Germany to Spain. The adaptive features were to a large degree possible because of the dialogue management architecture. However, the main focus of the dialogue management architecture lies not on these particular features but to provide dialogue designers with a method to model adaptive dialogue. The features that adaptive dialogue may have lie to a large extent in the hands of the domain designer. However, some points are quite special like the possibility to let the user choose what to do next in the dialogue, and the different forms of initiative goals. For evaluating these features with respect to user satisfaction we designed the scenario. In order to make a tentative evaluation of emotional adaptivity and expertise we included these features into the evaluation as well.

5.1 Portability

As designing an adaptive dialogue means more work than designing a non-adaptive dialogue, it is important to be able to reuse modeled dialogue structures for different modalities on the one hand and for defining similar parts of dialogues on the other hand. These two aspects of being portable have been realized with the dialogue widgets concept and with the externalisation of ressources which have already been explained. In order to provide a proof of concept for these advantages of the adaptive dialogue management architecture two scenarios have been realized. On the one hand a multimodal home cinema application which tutored the user on how to connect the cables of a new home cinema system. On the other hand a flight booking application was implemented which was speech controlled but had more adaptive parameters than the home cinema system, especially regarding emotional adaptivity.

5.1.1 Companion System - Home Cinema Application

The home cinema application was an interdisciplinary project for the presentation of research work done by several project areas like planning and scheduling, knowledge modelling, multimodal interaction and dialogue systems. It showed the cooperation of a planner, a knowledge base, a multimodal fusion component, a multimodal fission component and the mentioned dialogue manager. The result of this cooperation was a so-called companion system. The cooperation took place between the institute of artificial intelligence, the institute of media informatics and the institute of communication engineering.

The home cinema scenario included planning a part of the user's daily schedule and letting him interact with the planning component when it needed information. Providing the user with an individually adapted plan the planner considered context information about the user. The main part of the application consisted of an interactive tutorial to connect the freshly bought devices of a home cinema system which was realized using multiple modalities and devices. The dialogue as well as the user interface were adapted to information about the user context like his emotions, knowledge, position in the room, preferences and capabilities. The context information which were not available yet by the means of sensor information could be adjusted online manually.

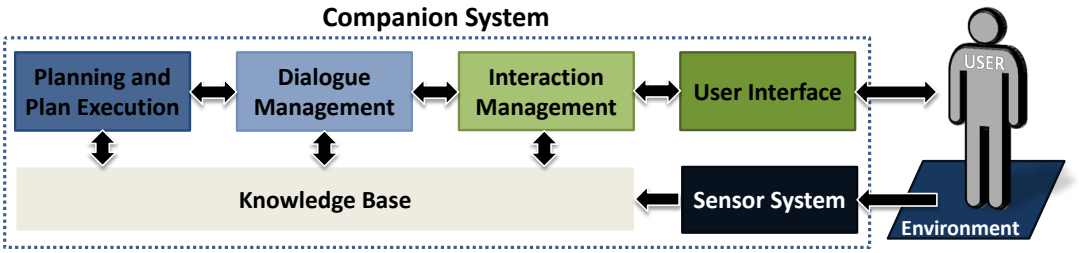

Figure 5.1: Architecture of the Companion System - The architecture used for the realization of the home cinema application as a companion system. The dialogue manager was used as a mediator between the planning component and the interaction management. Interaction management comprises the components of multimodal fission and fusion. The diagram is based on (Honold *et al.*, 2012, 2013). *Planning and Plan Execution, Dialogue Management* and *Interaction Management* components each exchange information with the *Knowledge Base* in order to adapt plan, dialogue and interaction to the information gathered by a sensor system. This information reflects assumptions about the user and his environment. The user interacts with the system with the help of a custom tailored interface.

The system showed the multimodal capabilities of the dialogue management architecture as well as its ability to adapt to environment parameters in the same way as emotional parameters. Furthermore it was integrated successfully into a large system architecture comprising different modules interacting with each other and therefore providing evidence of its flexibility.

5.1.2 Flight Booking Application

The flight booking application allowed the user to book a flight from Germany to Spain via speech interface. It allowed the user to specify start, destination, time and number of travellers while being adaptive to emotion and expertise. This application was designed on the one hand as a proof of concept for the mentioned concepts in adaptive dialogue management like user choice of dialogue order, mixed-initiative input, adaptivity to user expertise and user emotion. On the other hand the application served as a means to carry out a tentative user evaluation of these features.

5.2 User Test

Evaluating dialogue management with users is rather difficult because every user acts uniquely. For statistical evaluation the scenario has to be highly restricted. This may lead to aspects being lost. As the concepts developped during this thesis are quit abstract and general in order to carry out such an evaluation there must be a concrete interface which makes use of these general concepts. This has the immediate effect that parts of their generality are lost when specifying the realization of the interface. However an user test may give a prospective evaluation of some of the features.

Having users test the implemented features we used the flight booking application as a scenario in which several features that are possible with our architecture were evaluated. For comparing the user's rating of these features with the rating of their lack we implemented the system once with adaptive features and once without adaptive features. In this manner the user could use the system with the adaptive features and the system without adaptive features several times to judge their contribution to his satisfaction. The test had the form of three rounds in which the user should book a flight. Each round comprised the interaction with the adaptive and the non-adaptive system. In order to book the flight successfully the user needed to answer the questions of number of people flying, destination, arrival and departure time. After each round the user had to fill out a questionnaire about his satisfaction concerning certain features of the dialogue system. The study was conducted in a quasi Wizard of Oz experiment as the user interacted directly with a human who accepted his answers and entered them into the system. In this way speech recognition problems could be avoided. This experiment was conducted in cooperation with the institute of psychology and education.

5.2.1 Adaptivity to User Emotion

In order to test the adaptivity to user emotion we modeled the dialogue in a way that for any prompt the user had to answer there was an emotionally adaptive output from the system side. The system output was adapted to user emotions from all octants of the pleasure arousal dominance space. For each octant one sample emotion was used.

One question to be solved was how to evoke the emotions in the users which should test the system. After carefully assessing the arguments we chose a simple approach

in which the users had to remember a situation that corresponded to an emotion. We presented the user with several adjectives describing the particular octant of the PAD space. He then was asked to imagine a situation where he felt like the desired adjectives and was asked to imagine to be in this kind of emotional state for this round of the experiment. The adjectives were taken from Mehrabian (1998) and are listed in appedix C.1.3.

Then they were told to imagine to be in the described emotional state at that moment and to interact with the system. The system was set up to react according to that emotional state. In two rounds the user interacted with both the adaptive and the non-adaptive system. For each round they had another radomly selected emotional state that they should imagine to be in.

After each round the users compared the systems regarding the course of the dialogue respective to their emotional state on a seven point likert scale. An overview of the ratings is shown in figure 5.2 in a box-whiskers chart.

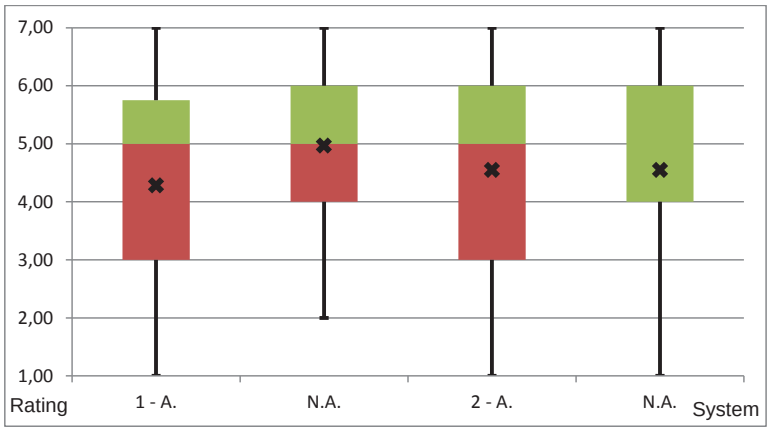

Figure 5.2: Ratings of Emotional Adaptivity - The users were to rate their satisfaction regarding the dialogue course with respect to their imagined emotion on a seven point likert scale for two rounds with each the adaptive and the non-adaptive system.

We also considered the ratings split by gender. An overview of this approach is shown in figure 5.3 and figure 5.4.

For male participants no significant results were found.

For female participants there was only one significant result. They disliked the reaction of the adaptive system more than the non-adaptive while they imagined their first emotions. With their second imagined emotions there was no significant result.

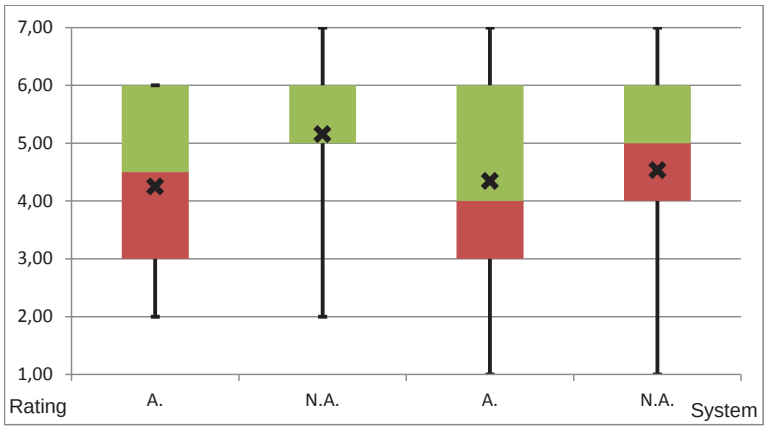

Figure 5.3: Ratings of Emotional Adaptivity for Females - The users were to rate their satisfaction regarding the dialogue course with respect to their imagined emotion on a seven point likert scale for two rounds with each the adaptive and the non-adaptive system. This chart shows the distribution for female participants.

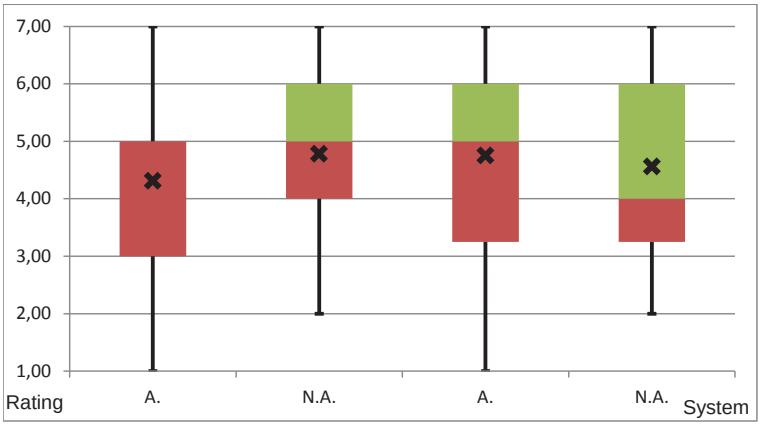

Figure 5.4: Ratings of Emotional Adaptivity for Males - The users were to rate their satisfaction regarding the dialogue course with respect to their imagined emotion on a seven point likert scale for two rounds with each the adaptive and the non-adaptive system. This chart shows the distribution for male participants.

5.2.2 Adaptivity to User Expertise

In order to find out if adaptivity to user expertise increases user satisfaction we reduced the information provided on how to book the flight from round one to round two in the adaptive system, while the non-adaptive system still provided the information which the user might have learned by now. After round two we asked the users about their satisfaction

Figure 5.5 provides an overview over the data collected in a box whiskers chart.

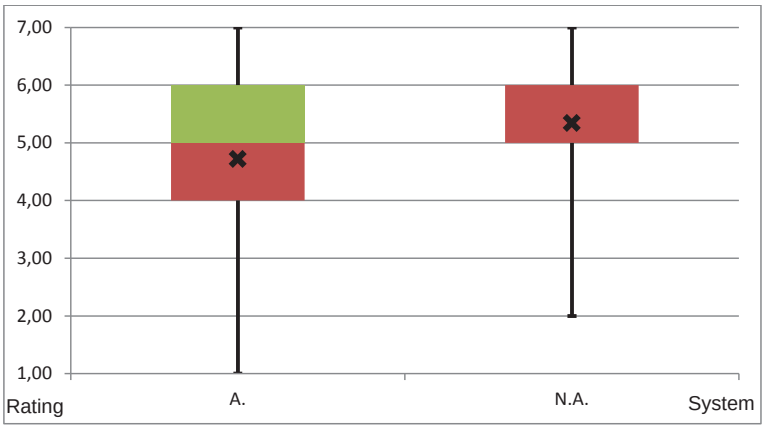

Figure 5.5: Ratings of Expertise - The users were to rate their satisfaction regarding the adaptivity to their expertise on a seven point likert scale after interacting with both adaptive and non-adaptive system after the second round.

The users did not rate the adaptive system significantly different from the non-adaptive.

Again we split the data into female and male to see if there was any difference. The results are shown in figures 5.6 and 5.7.

For expertise the female users showed no significant difference in the rating of the adaptive system against the non-adaptive system regarding expertise.

Male users, however, rated the non-adaptive system better than the adaptive.

5.2.3 Different Forms of Initiative

In order to find out how user initiative input might influence the satisfaction of the users we had them interact with both systems for a third time. Before interacting with them they were informed about the feature of user initiative input in the adaptive

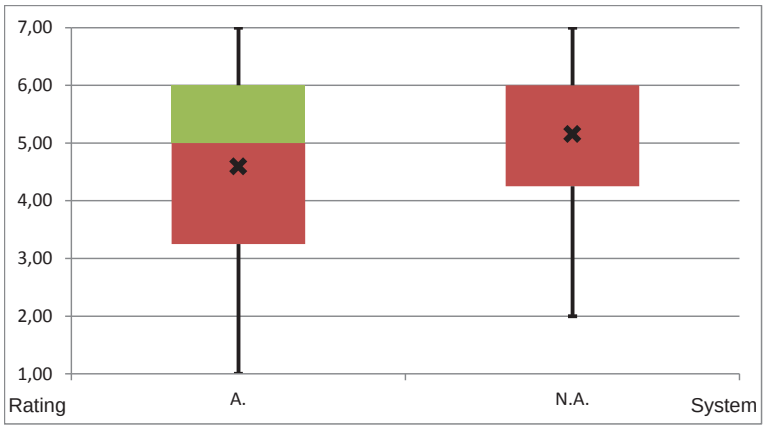

Figure 5.6: Ratings of Expertise for Males - The users were to rate their satisfaction regarding the adaptivity to their expertise on a seven point likert scale after interacting with both adaptive and non-adaptive system after the second round. This chart shows the distribution for female participants.

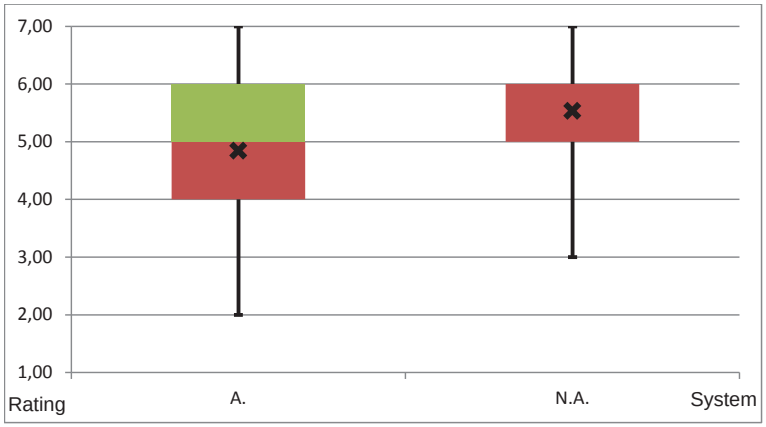

Figure 5.7: Ratings of Expertise for Females - The users were to rate their satisfaction regarding the adaptivity to their expertise on a seven point likert scale after interacting with both adaptive and non-adaptive system after the second round. This chart shows the distribution for female participants.

system. They had already learned how to use the systems during the first two rounds which made them able to finish their dialogue more quickly. After finishing the third round we asked them about their satisfaction with this feature.

An overview over the data collected is shown in figure 5.8.

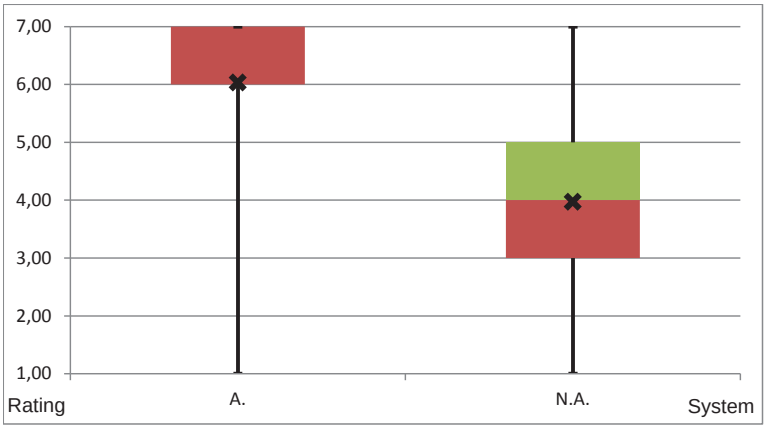

Figure 5.8: Ratings of Expertise - The users were to rate their satisfaction regarding the possibility of user initiative input in the adaptive system compared to the impossibility of user initiative input in the non-adaptive system after the third round.

We could show that the possibility of user initiative input increased the users' satisfaction significantly.

Splitting the data into female and male we wanted to find out possible differences. The results are shown in figures 5.9 and 5.10.

Female users rated the adaptive system significantly better than the non-adaptive system.

Male users also rated the adaptive system significantly better than the non-adaptive system.

5.2.4 Choice of Dialogue Order

Wishing to learn about the users' satisfaction to be able to choose what step to do next in the dialogue compared to not being able to do so we asked them after round 1 about this.

These results are shown in figure 5.11.

The users rated their satisfaction with the non-adaptive system significantly better than their satisfaction with the adaptive one.

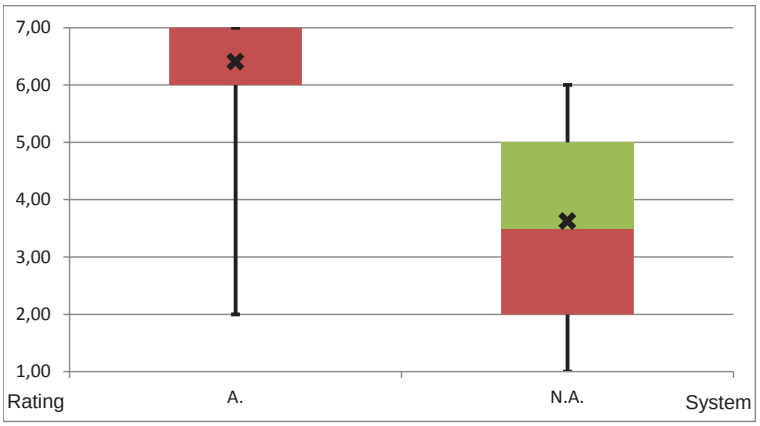

Figure 5.9: Ratings of Initiative for Males - The users were to rate their satisfaction regarding the possibility of user initiative input in the adaptive system compared to the impossibility of user initiative input in the non-adaptive system after the third round. This chart shows the distribution for female participants.

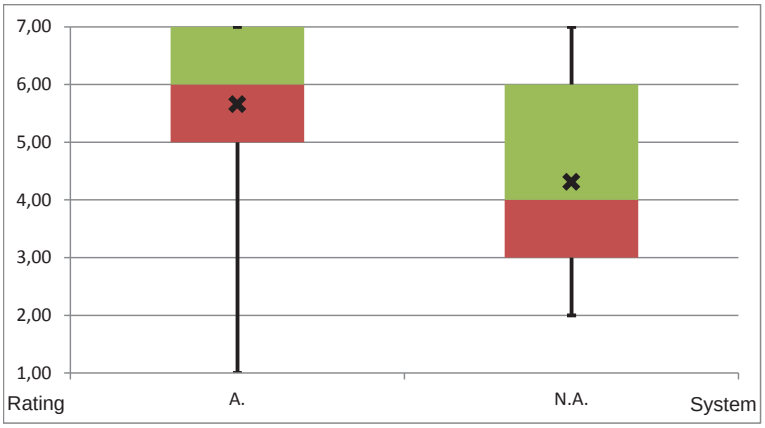

Figure 5.10: Ratings of Initiative for Females - The users were to rate their satisfaction regarding the possibility of user initiative input in the adaptive system compared to the impossibility of user initiative input in the non-adaptive system after the third round. This chart shows the distribution for female participants.

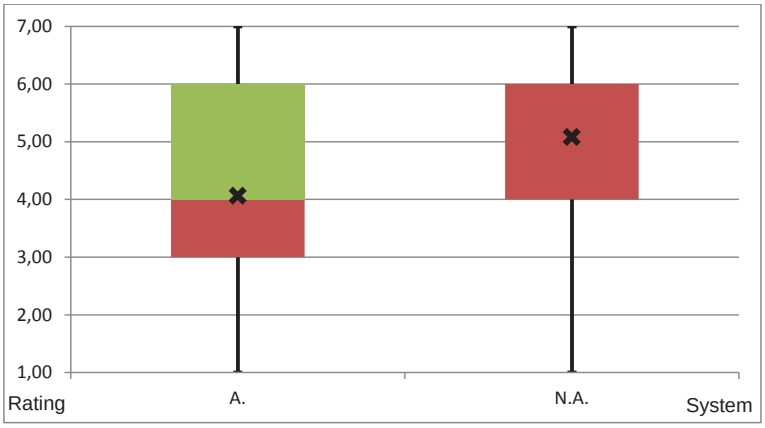

Figure 5.11: Ratings of Choice of Dialogue Order - The users were to rate their satisfaction regarding the possibility to choose how to continue the dialogue in the adaptive system compared to the impossibility of choosing how to continue the dialogue in the non-adaptive system after the first round.

Looking for differences we split the data by gender. The ratings can be seen in figures 5.12 and 5.13.

Female users rated the adaptive system significantly worse than the non-adaptive system.

Male users also rated the adaptive system significantly worse than the non-adaptive one.

5.2.5 Discussion of Results

As we have seen only the possibility of user initiative input made the users rate the adaptive system better than the non-adaptive system. For the features of emotional adaptivity and adaptivity to experience no significant statements concerning the users' satisfaction can be made. The feature of choosing the dialogue order even made the users rate the non-adaptive system better than the adaptive one. However, it is not likely that our results contradict well established human phenomena like the self determination theory of motivation which have been tested with larger empirical studies. We suppose that the particular reason for the rating was the quite small scenario where the user only wanted to finish the task of booking the flight was not the right domain to test this feature. For a larger domain this feature might prove more useful. In cases

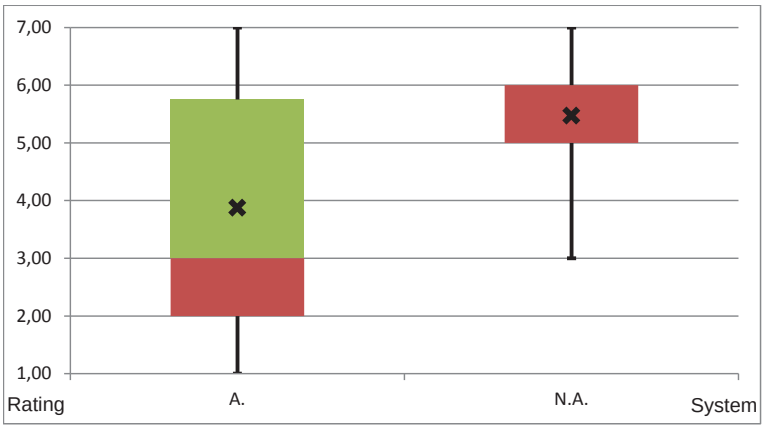

Figure 5.12: Ratings of Choice of Dialogue Order for Males - The users were to rate their satisfaction regarding the possibility to choose how to continue the dialogue in the adaptive system compared to the impossibility of choosing how to continue the dialogue in the non-adaptive system after the first round. This chart shows the distribution for female participants.

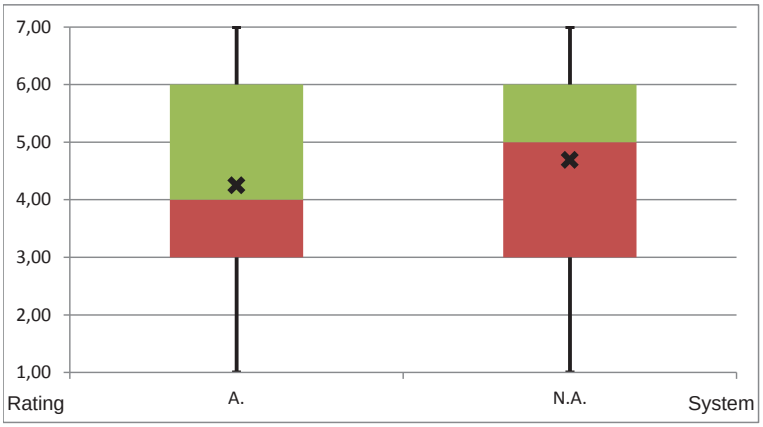

Figure 5.13: Ratings of Choice of Dialogue Order for Females - The users were to rate their satisfaction regarding the possibility to choose how to continue the dialogue in the adaptive system compared to the impossibility of choosing how to continue the dialogue in the non-adaptive system after the first round. This chart shows the distribution for female participants.

where the user has information about his schedule which the system cannot have it might help to let the user decide what he wants to do first and next.

Let us consider an example. An user has 2 appointments and for the second appointment he does need information which he only gets in the first appointment. The system does not know about this relation between the appointments. In this case it would be fatal not to let the user decide which appointment he wants to schedule first. The system might then schedule the appointments wrongly and the user could end up in the second appointment scheduled before the first appointment without knowing the needed information. He would waste his time. That is why we strongly believe that this feature may be of use and may in these cases also increase the user's satisfaction. However, in the modeled scenario there was no such relation between the tasks.

Regarding the feature of expertise the given domain might be also too small to make use of the full benefits of an expertise adaptive dialogue manager. For example, an expert might become bored if he hears the beginners instructions again and again while already having learnt them.

Considering the feature of emotional adaptivity it might prove useful to have an expert model the system reactions. These were based on our own thoughts. As has been mentioned earlier the field of how machines should react to human emotions has not been that widely investigated and it is difficult to find any best practices for this.

The main focus of this work was on designing an approach for dialogue management which makes these features possible and the specific form of adaptivity which might be applied in a domain scenario must be thought up by the dialogue designer. He may consult experts in the fields of emotion psychology and learning psychology for designing emotional adaptivity and expertise adaptivity.

5.3 Conclusion

In this chapter we have shown the portability of our system as well as the adaptivity functionality. For the adaptivity aspect we conducted a user study to learn about the user's satisfaction with one particular instance of adaptivity that is possible with the system. The user study proved that the aspect of user initiative input is useful for this instance because it increased the user's satisfaction. The aspect of choice of dialogue order was not useful in the small domain because it decreased the user's satisfaction.

However, based on the arguments already given by pschyological research we still believe that this feature will prove to be useful in other domains. Based on logical arguments presented we also can underline this. For the aspects of adaptivity to emotion and expertise we could not get any significant results from our study and further research of this topic might prove useful.

6

Conclusion and Future Directions

In this thesis we looked at the question how adaptive dialogues may be realized especially with the focus on emotion on the user side and with the focus on companions on the system side. The peculiarities of companion systems are that they have a lot of contextual information available about the user. This makes them an ideal system for adaptive interaction. In the field of dialogue management this means that the dialogue manager can make use of the context information in order to create a more desirable interaction experience for the user.

One particular point for adapting the interaction of such systems are user emotions. As user emotion detection is already being investigated thoroughly the topic of this thesis focused on how the dialogue manager can provide possibilities that enhance the creation of adaptive dialogues. Another part of the thesis was a tentative prod into the subject how exactly adaptivity to emotions may look like.

For this purpose a Wizard of Oz study was conducted in which different general possibilities of system reactions were investigated. In this study we not only gained insight in how the users react to the emotionally adaptive system but also were able to detect one important aspect that should guide the development of adaptive dialogue management systems. Providing the user with choice and freedom. Under this aspect we created a dialogue management architecture which not only allows multimodal interaction but also enables the user to arrange the dialogue as freely as possible.

In order to create an adaptive architecture we found it useful to seperate the different aspects of dialogue management into design, model, and control. We investigated how

adaptivity may influence each of these fields and how to respect the demands resulting from these influences.

6.1 Emotional Dialogue Strategies

Regarding the question of how to react to user emotions we investigated four different dialogue strategies: being thankful, praising the user, calming the user, and motivating the user. In a Wizard of Oz study we compared the strategies with respect to different user personalities. However, we were unable to find any personality traits which influence the preference of any of these four strategies. It has been shown that the users rate the system better if they can configure themselves which emotional dialogue strategies they would like. This conforms with the self-determination theory of motivation stating that the more a person feels to able to dermine his actions, the more motivated this person will be.

This also suggests that dialogue should provide many possiblilties for the user to interact freely with the system. That is why, we considered this aspect important for the design of our adaptive multimodal dialogue management architecture.

6.2 Adaptive Multimodal Dialogue Manager

Keeping in mind the aspects of user freedom in dialogue and adaptivity to user context our research in the dialogue management field especially adressed the question how to realize an adaptive system from the perspective of the designer as well as from the perspective of the dialogue control component which interprets the domain design at runtime. Based on these issues we created a dialogue model satisfying the needs of the domain designer and the requirements of dialogue control at the same time. In addition, the dialogue model provides multimodal functionality and portability.

6.2.1 Dialogue Model

To structure the semantics of the dialogue, the model provides goals as semantic containers and sub respectively next links which arrange the relations between these goals. Sub goal links specify that a dialogue goal may be further subdivided into several smaller semantic units. For example, having dinner may be subdivided into drinking

an aperitif, having the first course, having the second course, having the third course, having dessert and drinking a digestif. In the dialogue model this would be presented as a goal for having dinner which has a subgoal of drinking the aperitif with a next goal of the first course having a next goal of the second course and so on.

The access to the dialogue goals may be restricted through dialogue guards which model conditions imposed upon dialogue variables which have to be fulfilled so that the respective dialogue goal may be executed. For example the having dinner goal might only be executed if the user is hungry. In this manner conditions upon the user context as well as on the logical variables of the dialogue may influence the course of the dialogue. That is why the dialogue model practically sets no limits to the adaptive dialogues the designer may create.

It might also be possible that several dialogue goals may be executable at a certain time in the dialogue. In this case the user has the freedom to choose with which dialogue goal he wants to continue. This aspect of freedom seems to be crucial according to the self-determination theory of motivation. This theory states that people are more motivated to fulfill a task if they have the impression that their acting is free and self determined.

Another aspect of user freedom is reflected by different kinds of initiative which are possible with our dialogue model. The most important aspect is user initiative input which means that the user may leave the structure of the dialogue that the system presents to him through system initiative dialogue. He may ask questions or over answer. In this way using user initiative input he may break out of the system determined structure and act more freely. The fact, that both system initiative and user initiative are possible with the system makes it a mixed initiative architecture. Another kind of initiative is proactive system output. This may be used in different situations when contextual information makes it meaningful to act from the system side. A clumsy user might be offered help by the system proactively or a tired user might be offered some coffee.

By externalisation of ressources for the presentation information in so-called abstract information objects the created structures of the dialogue may be used for different modalities. In the abstract information objects presentation information for several modalities may be stored thus enabling the system to be used in a multimodal manner.

6.2.2 Dialogue Design

For the dialogue designer important aspects are the understandability of the dialogue model, its clarity and its structure. For him the internal workings of the dialogue control are not so important. However, because of the growing size and complexity arising from the adaptive features clarity and structure become more important. In order to be clear and allowing for a simple structure we developed a graphical editor which provides an xml document from which the dialogue model instance used by the dialogue control may be generated. This process fills java classes originating from an xsl transformation from a template document also created with the editor at the start of the dialogue.

The dialogue designer may arrange the objects of the dialogue model on a canvas and connect them via sub and next links in order to create a structure. He may partition the dialogue model into several files which can make the structure more clear. By encapsulating recurring dialogue structures into seperate files they may be reused in a different context by simply exchanging the referenced abstract information objects which specify the presentation information.

6.2.3 Dialogue Control

Dialogue control must fulfill two tasks. One is to interpret the structure that the dialogue designer specified for the dialogue. The other is to interpret context information available only at runtime and sometimes depending on the course of the dialogue so far.

For the first task a dialogue manager has been built which traverses the tree like structure of the dialogue model and decides which dialogue goals are under discussion depending on the current position in the dialogue. As proactive and user initiative goals are under discussion for all the structures which are lower in the tree these must be saved and managed. They must be available in case the user makes some user initiative input or the context variables change in a way that some proactive output should be provided.

For the second task a constraint solver is used which may interpret the dialogue variables which are initialized and determine variable conditions also for variables which are not initialized. In this way sensor information which is not present may be ignored

and executable dialogue goals may be determined. The constraint solver creates conditions for the goals which are under discussion which also reflect their guard contents. These conditions are then evaluated and possible variable configurations are found. Based on the possible variable configurations it is possible to determine the respective goals which are not only under discussion but also executable.

These two components work together in order to traverse the dialogue tree not only depending on its predefined structure, but also depending on the context variable information available at runtime. In this way adaptivity is achieved which is based on the domain model which was specified before runtime and based on the dialogue and context information only available at runtime.

6.3 Evaluation

For the evaluation part of the thesis two main categories were found to measure the functionality and quality of the dialogue manager. One category was portability and multimodal functionality and the other was adaptivity and user freedom. While portability and multimodal functionality are more technical aspects of the system which were addressed by proof of concept approaches, for the adaptivity and user freedom part a user study was conducted.

Considering portability and multimodality we implemented a demonstration system which worked in a multimodal context together with several other components. It directed the user to correctly connect a home cinema system. For the connections different cables were possible and the in- an output did not only work on several modalities but also on several devices even depending on the user's position in the room. Apart from the planning system which was used for the connection logic the system also had an explanation manager which could explain certain features of the system or peculiarities of the plan.

We also implemented a flight booking spoken dialogue system for booking a flight from germany to spain. This system was used with the help of a wizard who interpreted the user input in order to avoid speech recognition problems. In this system adaptive features considering user emotion, user expertise, user initiative and user choice of dialogue order were integrated which could not be integrated in the demonstration system.

While the demonstration system was a proof of concept for the multimodal functionality of the dialogue manager, the flight booking spoken dialogue system served as a user test platform for some adaptive features that are possible with the dialogue manager.

In a user test conducted with 64 persons we could show that the feature of user initiative input strongly improves user satisfaction. For the other features more tests seem to be neccessary especially in a larger domain to see if they may improve user satisfaction as well.

6.4 Future Directions

With respect to the different emotional dialogue strategies we propose to make each of them optional to the user in order to benefit from the self determination theory of motivation. In addition, as our evaluation has shown their success not to be dependent on the user's character but more on the situation we propose to investigate in which kind of situations these dialogue strategies are appropriate.

Considering the graphical editor for dialogue design there are some features like validation of application logic syntax and guard syntax which might be added. The syntax itself might also be refined in order to be more clear and more easily readable.

As the guard conditions in the dialogue goals are already used to find possible variable configurations a further logical step could be to use these variables for reasoning. In this way possible values for certain variables might be found by the constraint solver and proposed to the user as possible values. This might be achieved by tagging the variables for which the constraint solver should find a solution in a special way so that the dialogue manager knows that it should communicate the possible values determined by the constraint solver to the user. This change would affect the dialogue model as well as dialogue control.

Another way to configure the guard variables might be to use machine learning algorithms and corpora in order to find out optimal average values that the variables should have which regulate adaptivity of the dialogue. For this purpose POMDPs may be used to model the transition probabilities between different dialogue goals. In this way the guard variables might be adapted accordingly in their values.

The results from the user tests with the spoken dialogue flight booking system showed that user initiative may contribute to user satisfaction. However, the choice of dialogue order reduced user satisfaction which is supposedly due to the shortness of the test domain. In the short test domain the question of what to do first and so on was not relevant for the successful completion of the dialogue. The repeated question of what to do next made the dialogue longer. This interpretation does not contradict the self determination theory of motivation. It leads to the conclusion that it would be meaningful to leave the user the option to determine how to continue the dialogue not at any possible point in the dialogue but to model the dialogue in a way that at certain distinguished points the user has this possibility while at others it is not possible.

In order to get more meaningful results for adaptiveness towards user emotion and expertise it might be interesting to design bigger domains in which the user expertise is of more relevance than in the test domain. For emotional adaptivity our studies have shown that user satisfaction is not so much dependent on the user's personality traits but more on the situation that he is in. That is why in a bigger domain with more different situative configurations for the user there might be more insights to be found into how the emotional strategies of the system relate to user satisfaction.

As there is a lot of current research in the domain of human computer interaction especially in the area of companion systems and emotional adaptivity future results which may be integrated into the design of the dialogue domain will help to improve user satisfaction.

Appendices

Appendix A

Emotional Dialogue Strategies

For all utterances presented in the following sections, ”name” was replaced by the name of the user.

A.1 Utterances to Express Tankfulness towards the User

- Thanks for your concentrated cooperation, ”name”.
- Thank you for your active endeavor, ”name”.
- Thanks for your motivated cooperation, ”name”.
- Thank you that you are working so great, ”name”.
- Thank you that you are so focused on the task, ”name”.
- Thanks for the motivation shown, ”name”.
- Thank you for being so focused on the task, ”name”.
- Thank you for cooperating so well, ”name”.

A.2 Utterances to Praise the User in a General Way

- Your performance is very good, ”name”.
- Outstanding, ”name”.

- You do this perfectly, "name".
- You are getting better, "name".
- You're doing great, "name".
- You work quickly and effectively, "name".
- You do that very good so far, "name".
- Excellent performance so far, "name".

A.3 Utterances to Calm the User

- Stay calm, "name".
- Take it easy, "name".
- Don't get nervous, "name".
- Stay cool, "name".
- Don't let yourself get worked up, "name".
- Take your time, "name".
- In peace lies strength, "name".
- Be cool, "name".

A.4 Utterances to Motivate the User

- Show what you have inside, "name".
- Stay tuned, "name".
- Make an effort, "name".
- Be optimistic, "name".
- Don't give up, "name".

- You can do it, "name".
- Stay on task, "name".
- Show what you have got, "name".

Appendix B

Dialogue Model

The dialogue model contains two language formats which are parsed internally and which are defined in EBNF (ISO/IEC 14977).

B.1 Guard Language

The guards express variable conditions which are expressed by using the identifier of the variable utilized in the dialogue model and relations to numbers. A possible example may look like: pleasure >= 3 && arousal < 5.

EBNF syntax

guard = comparation, {("&&" | "||") comparation};

comparation = identifier, relation, number;

identifier = alphabetic character, { alphabetic character | digit } ;

number = digit, { digit };

relation = "==" | ">" | "<" | ">=" | "<=";

alphabetic character = "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z" | "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z" ;

digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" ;

white space = ? white space characters ? ;

all characters = ? all visible characters ? ;

B.2 Logical Statement Language

In the application logic or the effect of a dialogue goal any simple java program call can be called in the statement or respectively the logical statement field. Simple assignments of integer constants, and method results are possible. Variables refer to the variables used in the dialogue model. The method calls should refer to methods implemented in java.

EBNF syntax

assignment = variable, "=", (number | methodcall), ";";

methodcall = identifier, ".", identifier, "(", {identifier, {",", identifier}}, ")", ";";

variable = identifier, ".", identifier;

identifier = alphabetic character, { alphabetic character | digit };

number = digit, { digit };

alphabetic character = "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z" | "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z" ;

digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" ;

white space = ? white space characters ? ;

all characters = ? all visible characters ? ;

Appendix C

Evaluation

C.1 User Test

C.1.1 Introduction Text (translated from German)

In the following experiment you have the task to book an arbitrary flight with arbitrarily many persons from Germany to Spain. The system accepts your input via speech. As the system is not completely ready I will forward your speech input to the system.

The system exists in two different versions: system A is adaptive and system B is not adaptive.

A peculiar capability of system A is to change its behaviour if your emotion changes. For this purpose you are asked to put yourself into an emotional situation in order to compare system A with system B which cannot adapt to your emotional situation.

You can rate this and other features of system A afterwards in a questionnaire comparing it to system B.

For the rating you will do 3 rounds.

Before every round you will receive a short instruction, what to do.

C.1.2 Introduction for Emotional Rounds (translated from German)

Please try to remember a situation which is described by as many of the following words as possible in your emotional experience:

...

Imagine yourself now to be in this situation and this emotional condition and book the flight with the dialogue system.

C.1.3 Adjectives used for inducing emotions (translated from German)

We used for high pleasure, high arousal and high dominance:

- admired
- bold
- carfree
- excited
- might
- triumphant

For high pleasure, high arousal and low dominance:

- amazed
- fascinated
- grateful
- impressed
- loved
- respectful

For high pleasure, low arousal and high dominance:

- at ease
- comfortable
- relaxed
- satisfied
- secure

- unperturbed

For high pleasure, low arousal and low dominance:

- consoled
- docile
- protected
- reverent
- sleepy
- tranquilized

For low pleasure, high arousal and high dominance:

- angry
- catty
- defiant
- hostile
- insolent
- nasty

For low pleasure, high arousal and low dominance:

- aghast
- bewildered
- distressed
- in pain
- insecure
- upset

For low pleasure, low arousal and high dominance:

- disdainful
- indifferent
- selfish-uninterested
- uncaring
- unconcerend

For low pleasure, low arousal and low dominance:

- bored
- despairing
- fatigued
- lonely
- sad
- subdued

C.1.4 Questionnaires used for determining user satisfaction (translated from German)

All questionnaires could be answered on a 7 point likert scale (cf. (Likert, 1932) with 1 meaning not satisfied at all and 7 meaning completely satisfied (respectively 1 meaning worst and 7 meaning best for the question: "How well have you been able to put yourself into the emotional situation?").

Round 1

How satisfied have you been with the course of the dialogue with respect to your emotional situation for system A?

How satisfied have you been with the course of the dialogue with respect to your emotional situation for system B?

How well have you been able to put yourself into the emotional situation?

How satisfied have you been with the possibility to choose with which step to continue the dialogue for system A?

How satisfied have you been with not having to choose with which step to continue the dialogue for system B?

Round 2

How satisfied have you been with the dialogue being adapted to your experience in the usage of the system for system A?

How satisfied have you been with the dialogue not being adapted to your experience in the usage of the system for system B?

How satisfied have you been with the course of the dialogue with respect to your emotional situation for system A?

How satisfied have you been with the course of the dialogue with respect to your emotional situation for system B?

How well have you been able to put yourself into the emotional situation?

Introduction for Round 3

You can also answer mor than the presented question to system A. You can for example say at any time: "I want to travel with 10 persons from Berlin to Malaga on Tuesday."

For system B this is not possible.

Round 3

How satisfied have you been with being ablle to give all answers at once with system A?

How satisfied have you been with having to give each answer separately for system B?

C.1.5 Results of the Questionnaires

C.1.5.1 T Test

Satisfaction System A - System B	Paired Differences					T	df	Sig. (2-sided)
	Average	Standard Deviation	Standard Error of Average	95% Confidence Interval of Difference				
				Lower	Upper			
User Initiative Adaptivity	2,063	2,487	,311	1,441	2,684	6,634	63	,000
Emotion Round 1 Choice of	-,688	2,336	,292	-1,271	-,104	-2,355	63	,022
Dialogue Order Adaptivity	-1,016	2,941	,368	-1,750	-,281	-2,762	63	,008
Expertise Adaptivity	-,625	1,964	,245	-1,116	-,134	-2,546	63	,013
Emotion Round 2 Average of	0,000	2,649	,331	-,662	,662	0,000	63	1,000
all Features	-,344	1,697	,212	-,768	,080	-1,620	63	,110

C.1.5.2 T Test Females

Satisfaction System A - System B	Paired Differences					T	df	Sig. (2-sided)
	Average	Standard Deviation	Standard Error of Average	95% Confidence Interval of Difference				
				Lower	Upper			
User Initiative Adaptivity	2,781	2,379	,421	1,923	3,639	6,613	31	,000
Emotion Round 1 Choice of	-,906	2,161	,382	-1,685	-,127	-2,373	31	,024
Dialogue Order Adaptivity	-1,594	2,792	,494	-2,601	-,587	-3,229	31	,003
Expertise Adaptivity	-,563	2,368	,419	-1,416	,291	-1,344	31	,189
Emotion Round 2	-,188	2,867	,507	-1,221	,846	-,370	31	,714

C.1.5.3 T Test Males

Satisfaction System A - System B	Paired Differences					T	df	Sig. (2-sided)
	Average	Standard Deviation	Standard Error of Average	95% Confidence Interval of Difference				
				Lower	Upper			
User Initiative Adaptivity	1,344	2,418	,427	,472	2,215	3,144	31	,004
Emotion Round 1	-,469	2,514	,444	-1,375	,438	-1,055	31	,300
Choice of Dialogue Order Adaptivity	-,438	3,015	,533	-1,525	,650	-,821	31	,418
Expertise Adaptivity	-,688	1,491	,263	-1,225	-,150	-2,609	31	,014
Emotion Round 2	,188	2,442	,432	-,693	1,068	,434	31	,667

References

Abler, B. & H., K. (2009). Emotion Regulation Questionnaire - Eine deutschsprachige Fassung des ERQ von Gross und John. *Diagnostica*, **55**.

Ai, H., Litman, D., Forbes-Riley, K., Rotaru, M., Tetreault, J. & Purandare, A. (2006). Using system and user performance features to improve emotion detection in spoken tutoring dialogs. *Proceedings of Interspeech*, 797–800.

Allen, J., Byron, D., Dzikovska, M., Ferguson, G. & Galescu, L. (2001). Toward conversational human-computer interaction. *AI Mag.*, **22**, 27–37.

Allen, J.F. (1979). *A plan-based approach to speech act recognition*. Ph.D. thesis, University of Toronto, aAI0532011.

Allgayer, J., Harbusch, K., Kobsa, A., Reddig, C., Reithinger, N. & Schmauks, D. (1989). Xtra: a natural-language access system to expert systems. *International Journal of Man-Machine Studies*, **31**, 161 – 195.

André, E., Rehm, M., Minker, W. & Bühler, D. (2004). Endowing spoken language dialogue systems with emotional intelligence. In E. André, L. Dybkjaer, W. Minker & P. Heisterkamp, eds., *Affective Dialogue Systems*, vol. 3068 of *Lecture Notes in Computer Science*, 178–187, Springer Berlin Heidelberg.

Arnold, M.B. (1960). *Emotion and Personality. Vols. 1, 2*. Columbia University Press.

Austermann, A., Esau, N., Kleinjohann, L. & Kleinjohann, B. (2005). Fuzzy emotion recognition in natural speech dialogue. In *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on*, 317 – 322.

Averill, J. (1980). A constructivist view of emotion. *Emotion: Theory, Research and Experience*, **1**, 305–339.

Barbosa, P. (2009). Detecting changes in speech expressiveness in participants of a radio program. In *Tenth Annual Conference of the International Speech Communication Association*.

Barra-chicote, R., Fern, F., Lutfi, S., Lucas-cuesta, J.M., Macias-guarasa, J., Montero, J.M., San-segundo, R. & Pardo, J.M. (2009). Acoustic emotion recognition using dynamic bayesian networks and multi-space distributions. In *Proceedings of the 10th Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

Bartak, R. (1999). Constraint programming: In pursuit of the holy grail. vol. 4 of *Week of Doctoral Students*, 555–564, MatFyzPress.

Bartneck, C. (2002). Integrating the occ model of emotions in embodied characters. *Workshop on Virtual Conversational Characters*.

Batliner, A., Fischer, K., Huber, R., Spilker, J. & Nöth, E. (2003a). How to find trouble in communication. *Speech Communication*, **40**, 117 – 143.

Batliner, A., Zeissler, V., Frank, C., Adelhardt, J., Shi, R. & Nöth, E. (2003b). We are not amused-but how do you know? user states in a multi-modal dialogue system. *Proceedings of EUROSPEECH*, 733–736.

Batliner, A., Hacker, C., Steidl, S., Nöth, E. & Haas, J. (2004). From emotion to interaction: Lessons from real human-machine-dialogues. In E. André, L. Dybkjaer, W. Minker & P. Heisterkamp, eds., *Affective Dialogue Systems*, vol. 3068 of *Lecture Notes in Computer Science*, 1–12, Springer Berlin Heidelberg.

Batliner, A., Steidl, S., Hacker, C., Nöth, E. & Niemann, H. (2005). Tales of tuning–prototyping for automatic classification of emotional user states. *Proc. 9th Eurospeech-Interspeech 2005*, 489–492.

Batliner, A., Steidl, S., Hacker, C. & Nöth, E. (2008). Private emotions versus social interaction: a data-driven approach towards analysing emotion in speech. *User Modeling and User-Adapted Interaction*, **18**, 175–206.

Baumeister, R.F., Hutton, D.G. & Cairns, K.J. (1990). Negative effects of praise on skilled performance. *Basic and Applied Social Psychology*, **11**, 131–148.

Becher, M. (2009). *XML - DTD, XML-Schema, XPath, XQuery, XSLT, XSL-FO, SAX, DOM*. W3L Verlag.

Bertrand, G., Nothdurft, F., Honold, F. & Schüssel, F. (2011). Caligraphi - creation of adaptive dialogues using a graphical interface. In *35th Annual IEEE Computer Software and Applications Conference*, COMPSAC, 393–400, CPS, IEEE.

Billsus, D. & Pazzani, M. (2000). User modeling for adaptive news access. *User Modeling and User-Adapted Interaction*, **10**, 147–180.

Bitouk, D., Verma, R. & Nenkova, A. (2010). Class-level spectral features for emotion recognition. *Speech Communication*, **52**, 613 – 625.

Blank, K. (1996). *Benutzermodellierung für adaptive interaktive Systeme: Architektur, Methoden, Werkzeuge und Anwendungen*. Ph.D. thesis, University of Stuttgart.

Böck, R., Gluge, S., Wendemuth, A., Limbrecht, K., Walter, S., Hrabal, D. & Traue, H. (2012). Intraindividual and interindividual multimodal emotion analyses in human-machine-interaction. In *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2012 IEEE International Multi-Disciplinary Conference on*, 59 – 64.

Borkenau, P. & Ostendorf, F. (1993). *NEO-Fünf-Faktoren Inventar (NEO-FFI); nach Costa und McCrae*. Hogrefe.

REFERENCES

Brajnik, G. & Tasso, C. (1994). A shell for developing non-monotonic user modeling systems. *International Journal of Human-Computer Studies*, **40**, 31 – 62.

Brave, S., Nass, C. & Hutchinson, K. (2005). Computers that care: investigating the effects of orientation of emotion exhibited by an embodied computer agent. *International Journal of Human-Computer Studies*, **62**, 161 – 178, subtle expressivity for characters and robots.

Breese, J.S., Heckerman, D. & Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, UAI'98, 43–52, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Brown, P., Levinson, S.C. & Gumperz, J.J. (1987). *Politeness: Some Universals in language Use*. Cambridge University Press.

Buchanan, B.G. & Shortliffe, E.H. (1984). *Rule Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project (The Addison-Wesley series in artificial intelligence)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Bui, T.H., Poel, M., Nijolt, A. & Zwiers, J. (2009). A tractable hybrid ddn-pomdp approach to affective dialogue modeling for probabilistic frame-based dialogue systems. *Natural Language Engineering*, **15**, 273–307.

Burkhardt, F., van Ballegooy, M., Englert, R. & Huber, R. (2005). An emotion-aware voice portal. *Proc. Electronic Speech Signal Processing ESSP*, 123–131.

Busso, C., Deng, Z., Yildirim, S., Bulut, M., Lee, C.M., Kazemzadeh, A., Lee, S., Neumann, U. & Narayanan, S. (2004). Analysis of emotion recognition using facial expressions, speech and multimodal information. In *Proceedings of the 6th international conference on Multimodal interfaces*, ICMI '04, 205–211, ACM, New York, NY, USA.

Cannon, W.B. (1927). The james-lange theory of emotions: A critical examination and an alternative theory. *The American Journal of Psychology*, **39**, 106–124.

Caridakis, G., Malatesta, L., Kessous, L., Amir, N., Raouzaiou, A. & Karpouzis, K. (2006). Modeling naturalistic affective states via facial and vocal expressions recognition. In *Proceedings of the 8th international conference on Multimodal interfaces*, 146–154, ACM.

Carver, C.S. & White, T.L. (1994). Behavioral inhibition, behavioral activation, and affective responses to impending reward and punishment: The bis/bas scales. *Journal of Personality and Social Psychology*, **67**, 319–333.

Casale, S., Russo, A., Scebba, G. & Serrano, S. (2008). Speech emotion classification using machine learning algorithms. In *Semantic Computing, 2008 IEEE International Conference on*, 158 –165.

Castellano, G., Kessous, L. & Caridakis, G. (2008). Emotion recognition through multiple modalities: Face, body gesture, speech. In C. Peter & R. Beale, eds., *Affect and Emotion in Human-Computer Interaction*, vol. 4868 of *Lecture Notes in Computer Science*, 92–103, Springer Berlin Heidelberg.

Cavalluzzi, A., Carolis, B., Carofiglio, V. & Grassano, G. (2003). Emotional dialogs with an embodied agent. In P. Brusilovsky, A. Corbett & F. Rosis, eds., *User Modeling 2003*, vol. 2702 of *Lecture Notes in Computer Science*, 86–95, Springer Berlin Heidelberg.

Chandler, P. & Sweller, J. (1991). Cognitive load theory and the format of instruction. *Cognition and Instruction*, **8**, 293–332.

Chen, K., Yue, G., Yu, F., Shen, Y. & Zhu, A. (2007). Research on speech emotion recognition system in e-learning. In Y. Shi, G. Albada, J. Dongarra & P. Sloot, eds., *Computational Science - ICCS 2007*, vol. 4489 of *Lecture Notes in Computer Science*, 555–558, Springer Berlin Heidelberg.

Cho, J., Kato, S. & Itoh, H. (2007). Bayesian-based inference of dialogist's emotion for sensitivity robots. In *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on*, 792 –797.

Cichosz, J. & Slot, K. (2007). Emotion recognition in speech signal using emotion-extracting binary decision trees. In *Proceedings of the 2nd International Conference on Affective Computing and Intelligent Interaction*.

Clavel, C., Devillers, L., Richard, G., Vasilexcu, I. & Ehrette, T. (2007). Detection and analysis of abnormal situations through fear-type acoustic manifestations. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4, IV–21 –IV–24.

Cohen, P.R. & Perrault, C.R. (1979). Elements of a plan-based theory of speech acts. *Cognitive Science*, **3**, 177 – 212.

Dahlbäck, N., Flycht-eriksson, A., Jonsson, A. & Qvarfordt, P. (1999). An architecture for multi-modal natural dialogue systems. In *In Proceedings of ESCA Tutorial and Research Workshop (ETRW) on Interactive Dialogue in Multi-Modal Systems*.

Darwin, C. (1972). *The Expression of the Emotions in Man and Animals*. John Murray.

De Carolis, B., Pelachaud, C., Poggi, I. & de Rosis, F. (2001). Behavior planning for a reflexive agent. *International Joint Conference on Artificial Intelligence*, **17**, 1059–1066.

Deci, E.L. & Ryan, R.M. (1985). *Intrinsic Motivation and Self-Determination in Human Behavior*. Springer.

Devillers, L., Vidrascu, L. & Lamel, L. (2005). 2005 special issue: Challenges in real-life emotion annotation and machine learning based detection. *Neural Netw.*, **18**, 407–422.

Duffy, E. (1941). An explanation of "emotional" phenomena without the use of the concept "emotion.". *Journal of General Psychology*, **25**, 283–293.

Ehrenthal, J., Dinger, U., Lamla, A., Funken, B. & Schauenburg, H. (2009). Evaluation der deutschsprachigen Version des Bindungsfragebogens [Experiences in Close Relationship - Revised]. *Psychotherapie, Psychosomatik, Medizinische Psychologie*, **59**, 215–223.

Ekman, P., Friesen, W.V. & Ellsworth, P. (1972). *Emotion in the human face: Guidelines for research and an integration of findings.*. Oxford.

Emmons, R.A. (2004). *The Psychology of Gratitude*. Oxford University Press.

REFERENCES

Eyben, F., Wollmer, M. & Schuller, B. (2009). Openear - introducing the munich open-source emotion and affect recognition toolkit. In *Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on*, 1 –6.

Fernández, A.J., Hortalá-González, M.T., Sáenz-Pérez, F. & del Vado Vírseda, R. (2006). Constraint functional logic programming over finite domains. *CoRR*, **abs/cs/0601071**.

Fernandez, R. & Picard, R.W. (2003). Modeling drivers' speech under stress. *Speech Communication*, **40**, 145 – 159.

Fernandez, R. & Picard, R.W. (2005). Classical and novel discriminant features for affect recognition from speech. In *Proceedings of the 9th European Conference on Speech Communication and Technology*.

Fersini, E., Messina, E. & Archetti, F. (2012). Emotional states in judicial courtrooms: An experimental investigation. *Speech Communication*, **54**, 11 – 22.

Finin, T. & Drager, D. (1986). Gums: a general user modeling system. In *Proceedings of the workshop on Strategic computing natural language*, HLT '86, 224–230, Association for Computational Linguistics, Stroudsburg, PA, USA.

Fink, J. (1999). Transactional consistency in user modeling systems. In *PROCEEDINGS OF THE SEVENTH INTERNATIONAL CONFERENCE*, 191–200, Springer-Verlag.

Fink, J. & Kobsa, A. (2000). A review and analysis of commercial user modeling servers for personalization on the world wide web. *User Modeling and User-Adapted Interaction*, **10**, 209–249.

Fortenbaugh, W.W. (1975). *Aristotle on emotion: A contribution to philosophical psychology, rhetoric, poetics, politics, and ethics*. Barnes & Noble Books.

Fragopanagos, N. & Taylor, J.G. (2005). Emotion recognition in human-computer interaction. *Neural Networks*, **18**, 389–405.

Frey, D. & Irle, M. (2002). *Theorien der Sozialpsychologie*. Huber.

Gajšek, R., Štruc, V., Dobrišek, S. & Mihelič, F. (2009). Emotion recognition using linear transformations in combination with video. In *Tenth Annual Conference of the International Speech Communication Association*.

Giannakopoulos, T., Pikrakis, A. & Theodoridis, S. (2009). A dimensional approach to emotion recognition of speech from movies. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, 65 –68.

Go, H.J., Kwak, K.C., Lee, D.J. & Chun, M.G. (2003). Emotion recognition from the facial image and speech signal. In *SICE 2003 Annual Conference*, vol. 3, 2890 –2895 Vol.3.

Gollwitzer, M. & Schmitt, M. (2009). *Sozialpsychologie kompakt*. Beltz.

Gratch, J. & Marsella, S. (2004). A domain-independent framework for modeling emotion. *Cognitive Systems Research*, **5**, 269 – 306.

Grimm, M., Kroschel, K., Harris, H., Nass, C., Schuller, B., Rigoll, G. & Moosmayr, T. (2007a). On the necessity and feasibility of detecting a driver's emotional state while driving. In A. Paiva, R. Prada & R. Picard, eds., *Affective Computing and Intelligent Interaction*, vol. 4738 of *Lecture Notes in Computer Science*, 126–138, Springer Berlin Heidelberg.

Grimm, M., Kroschel, K., Mower, E. & Narayanan, S. (2007b). Primitives-based evaluation and estimation of emotions in speech. *Speech Communication*, **49**, 787 – 800, ¡ce:title¿Intrinsic Speech Variations¡/ce:title¿.

Haag, A., Goronzy, S., Schaich, P. & Williams, J. (2004). Emotion recognition using bio-sensors: First steps towards an automatic system. In E. André, L. Dybkjaer, W. Minker & P. Heisterkamp, eds., *Affective Dialogue Systems*, vol. 3068 of *Lecture Notes in Computer Science*, 36–48, Springer Berlin Heidelberg.

Harre, R. (1986). *The Social Construction of Emotions*. John Wiley & Sons, Limited.

Hassan, A. & Damper, R.I. (2009). Emotion recognition from speech using extended feature selection and a simple classifier. In *Proceeedings of the 10th Annual Conference of the International Speech Communication Association (Interspeech)*.

Heinroth, T. (2012). *Model-Driven Adaptation for Spoken Dialogues in Intelligent Environments*. Ph.D. thesis, Ulm University.

Heinroth, T., Denich, D. & Schmitt, A. (2010). Owlspeak - adaptive spoken dialogue within intelligent environments. In *8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 666 – 671.

Heinroth, T., Kameas, A., Pruvost, G., Seremeti, L., Bellik, Y. & Minker, W. (2011). Human-Computer Interaction in Next Generation Ambient Intelligent Environments. *Intelligent Decision Technologies - Special Issue on Knowledge-based environments and services in HCI*, **5**.

Herlocker, J.L., Konstan, J.A., Borchers, A. & Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, 230–237, ACM, New York, NY, USA.

Hilgard, E.R. (1980). The trilogy of mind: Cognition, affection, and conation. *Journal of the History of the Behavioral Sciences*, **16**, 107–117.

Hillard, D., Ostendorf, M. & Shriberg, E. (2003). Detection of agreement vs. disagreement in meetings: training with unlabeled data. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume of the Proceedings of HLT-NAACL 2003–short papers - Volume 2*, NAACL-Short '03, 34–36, Association for Computational Linguistics, Stroudsburg, PA, USA.

Hoch, S., Althoff, F., McGlaun, G. & Rigoll, G. (2005). Bimodal fusion of emotional data in an automotive environment. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, vol. 2, ii/1085 – ii/1088 Vol. 2.

REFERENCES

Holzapfel, H. & Fuegen, C. (2002). Integrating emotional cues into a framework for dialogue management. In *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces*, ICMI '02, 141–, IEEE Computer Society, Washington, DC, USA.

Hone, K. (2006). Emphatic agents to reduce user frustration: The effects of varying agent characteristics. *Interacting With Computers*, **18**, 227–245.

Honold, F., Schüssel, F. & Weber, M. (2012). Adaptive probabilistic fission for multimodal systems. In *OzCHI '12: Proceedings of the 24th Australian Computer-Human Interaction Conference*, ACM, Melbourne, Australia.

Honold, F., Schüssel, F., Weber, M., Nothdurft, F., Bertrand, G. & Minker, W. (2013). Context models for adaptive dialogs and multimodal interaction. Intelligent Environments.

Hrabal, D., Rukavina, S., Limbrecht, K., Gruss, S., Walter, S., Hrabal, V. & Traue, H.C. (2012). Emotion identifcation and modelling on the basis of paired physiological data features for companion systems. In *MathMod*.

Huang, X., McCalla, G., Greer, J. & Neufeld, E. (1991). Revising deductive knowledge and stereotypical knowledge in a student model. *User Modeling and User-Adapted Interaction*, **1**, 87–115.

Ijima, Y., Tachibana, M., Nose, T. & Kobayashi, T. (2009). Emotional speech recognition based on style estimation and adaptation with multiple-regression hmm. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, 4157 –4160.

Izard, C.E. (1994). Innate and universal facial expressions: Evidence from developmental and cross-cultural research. *Psychological Bulletin*, **115**, 288–299.

Jaksic, N., Branco, P., Stephenson, P. & Encarnação, L.M. (2006). The effectiveness of social agents in reducing user frustration. In *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*, 917–922, ACM, New York, NY, USA.

James, W. (1884). What is an emotion? *Mind*, **34**, 188–205.

Jones, C. & Jonsson, I.M. (2008). Using paralinguistic cues in speech to recognise emotions in older car drivers. In C. Peter & R. Beale, eds., *Affect and Emotion in Human-Computer Interaction*, vol. 4868 of *Lecture Notes in Computer Science*, 229–240, Springer Berlin Heidelberg.

Jones, C.M. & Jonsson, I.M. (2005). Automatic recognition of affective cues in the speech of car drivers to allow appropriate responses. In *Proceedings of the 17th Australia conference on Computer-Human Interaction: Citizens Online: Considerations for Today and the Future*, OZCHI '05, 1–10, Computer-Human Interaction Special Interest Group (CHISIG) of Australia, Narrabundah, Australia, Australia.

Jussien, N., Rochart, G. & Lorca, X. (2008). The choco constraint programming solver. *CPAIOR'08 workshop on Open-Source Software for Integer and Contraint Programming (OSSICP'08)*.

Kass, R. (1991). Building a user model implicitly from a cooperative advisory dialog. *User Modeling and User-Adapted Interaction*, **1**, 203–258, 10.1007/BF00141081.

Kay, J. (1995). The um toolkit for reusable, long term user models. *User Modeling and User-Adapted Interaction*, **4**, 149–196.

Kim, E.H., Hyun, K.H., Kim, S.H. & Kwak, Y.K. (2009). Improved emotion recognition with a novel speaker-independent feature. *Mechatronics, IEEE/ASME Transactions on*, **14**, 317 –325.

Kim, J. (2007). Bimodal emotion recognition using speech and physiological changes. In *Robust Speech Recognition and Understanding*, chap. 15, 265–280, I-Tech Education and Publishing.

Kim, J. & Andre, E. (2008). Emotion recognition based on physiological changes in music listening. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **30**, 2067 –2083.

Kim, K., Bang, S. & Kim, S. (2004). Emotion recognition system using short-term monitoring of physiological signals. *Medical and Biological Engineering and Computing*, **42**, 419–427.

Klein, J., Moon, Y. & Picard, R. (2002). This computer responds to user frustration:: Theory, design, and results. *Interacting with Computers*, **14**, 119–140.

Kobsa, A. (1990). Modeling the user's conceptual knowledge in bgp-ms, a user modeling shell system1. *Computational Intelligence*, **6**, 193–208.

Kobsa, A. (1994). User modeling and user-adapted interaction. In *Conference Companion on Human Factors in Computing Systems*, CHI '94, 415–416, ACM, New York, NY, USA.

Kobsa, A. (2001). Generic user modeling systems. *User Modeling and User-Adapted Interaction*, **11**, 49–63.

Koestner, R., Zuckerman, M. & Koestner, J. (1987). Praise, involvement, and intrinsic motivation. *Journal of Personality and Social Psychology*, **53**, 383–390.

Kono, Y., Ikeda, M. & Mizoguchi, R. (1994). Themis: a nonmonotonic inductive student modeling system. *Journal of Artificial Intelligence in Education*, **5**, 371–371.

Kwon, O.W., Chan, K., Hao, J. & T.-W., L. (2003). Emotion recognition by speech signals. In *Proceedings of the 8th European Conference on Speech Communication and Technology (INTERSPEECH)*.

Larsson, S. & Traum, D.R. (2000). Information state and dialogue management in the trindi dialogue move engine toolkit. *Natural Language Engineering*, **6**, 323–340.

Laurent, J.P., Ayel, J., Thome, F. & Ziebelin, D. (1984). Comparative evaluation of three expert system development tools: Kee, knowledge craft, art. *The Knowledge Engineering Review*, **1**, 18–29.

Lazarus, R.S. (1968). Emotions and adaptation: Conceptual and empirical relations. *Nebraska Symposium on Motivation*, **16**, 175–266.

Lee, C.C., Mower, E., Busso, C., Lee, S. & Narayanan, S. (2011). Emotion recognition using a hierarchical binary decision tree approach. *Speech Communication*, **53**, 1162 – 1171, ¡ce:title¿Sensing Emotion and Affect - Facing Realism in Speech Processing¡/ce:title¿.

Lee, C.M. & Narayanan, S. (2005). Toward detecting emotions in spoken dialogs. *Speech and Audio Processing, IEEE Transactions on*, **13**, 293 – 303.

Lee, C.M., Yildirim, S., Bulut, M., Kazemzadeh, A., Busso, C., Deng, Z., Lee, S. & Narayanan, S. (2004). Emotion recognition based on phoneme classes. In *Proceedings of ICSLP*.

Lemon, O., Bracy, A., Gruenstein, A. & Peters, S. (2001). The witas multi-modal dialogue system i. In *EUROSPEECH*, 1559–1562.

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology*, **22**, 55.

Lin, Y.P., Wang, C.H., Jung, T.P., Wu, T.L., Jeng, S.K., Duann, J.R. & Chen, J.H. (2010). Eeg-based emotion recognition in music listening. *Biomedical Engineering, IEEE Transactions on*, **57**, 1798 –1806.

Litman, D., Rotaru, M. & Nicholas, G. (2009). Classifying turn-level uncertainty using word-level prosody. *Proceedings of Interspeech*, **9**.

Litman, D.J. & Forbes-Riley, K. (2004). Predicting student emotions in computer-human tutoring dialogues. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Association for Computational Linguistics, Stroudsburg, PA, USA.

Lugger, M. & Yang, B. (2008). Cascaded emotion classification via psychological emotion dimensions using a large set of voice quality parameters. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, 4945 –4948.

LuperFoy, S., Loehr, D., Duff, D., Miller, K., Reeder, F. & Harper, L. (1998). An architecture for dialogue management, context tracking, and pragmatic adaptation in spoken dialogue systems. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 2*, ACL '98, 794–801, Association for Computational Linguistics, Stroudsburg, PA, USA.

Machado, I., Martins, A., Martins, R. & Paiva, A. (1999). One for all and all in one - a learner modelling server in a multi-agent platform. In *In: Proceedings of the Seventh International Conference on User Modelling*, 211–221, Springer-Verlag.

Martin, D.L., Cheyer, A.J. & Moran, D.B. (1999). The open agent archictecture: A framework for building distributed software systems. *Applied Artificial Intelligence*, **13**, 91–128(38).

McTear, M.F. (2002). Spoken dialogue technology: enabling the conversational user interface. *ACM Comput. Surv.*, **34**, 90–169.

Mehrabian, A. (1996). Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament. *Current Psychology*, **14**, 261–292.

Mehrabian, A. (1998). Manual for a comprehensive system of measures of emotional states: The pad model. *Available from Albert Mehrabian*, **1130**, 14–16.

Meudt, S. & Schwenker, F. (2012). On instance selection in audio based emotion recognition. In N. Mana, F. Schwenker & E. Trentin, eds., *Artificial Neural Networks in Pattern Recognition*, vol. 7477 of *Lecture Notes in Computer Science*, 186–192, Springer Berlin Heidelberg.

Mower, E., Mataric, M. & Narayanan, S. (2009). Evaluating evaluators: A case study in understanding the benefits and pitfalls of multievaluator modeling. *10th Annual Conference of the International Speech Communication Association (Interspeech)*.

Nasoz, F., Alvarez, K., Lisetti, C. & Finkelstein, N. (2004). Emotion recognition from physiological signals using wireless sensors for presence technologies. *Cognition, Technology & Work*, **6**, 4–14.

Neiberg, D. & Elenius, K. (2008). Automatic recognition of anger in spontaneous speech. In *Ninth Annual Conference of the International Speech Communication Association*.

Neiberg, D., Elenius, K. & Laskowski, K. (2006). Emotion recognition in spontaneous speech using gmms. In *Ninth International Conference on Spoken Language Processing*.

Nicholas, G., Rotaru, M. & Litman, D. (2006). Exploiting word-level features for emotion prediction. In *Spoken Language Technology Workshop, 2006. IEEE*, 110–113, IEEE.

Nigay, L. & Coutaz, J. (1995). A generic platform for addressing the multimodal challenge. In *Proceedings of the SIGCHI conference on Human factors in computing systems*.

Nothdurft, F., Bertrand, G., Heinroth, T. & Minker, W. (2010). Geedi - guards for emotional and explanatory dialogues. In *6th International Conference on Intelligent Environments (IE'10)*.

Nwe, T.L., Foo, S.W. & Silva, L.C.D. (2003). Speech emotion recognition using hidden markov models. *Speech Communication*, **41**, 603 – 623.

Nydahl, L.O. (2004). *Wie die Dinge sind*. Knaur.

O'Neill, I., Hanna, P., Liu, X. & McTear, M. (2003). The queen's communicator: An object-oriented dialogue manager. In *EUROSPEECH*, 593–596.

Ortony, A. & Clore, A., Gerald L.and Collins (1988). *The cognitive structure of emotions.*. Cambridge University Press.

Orwant, J. (1994). Heterogeneous learning in the doppelgänger user modeling system. *User Modeling and User-Adapted Interaction*, **4**, 107–130.

Osherenko, A., Andre, E. & Vogt, T. (2009). Affect sensing in speech: Studying fusion of linguistic and acoustic features. In *Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on*, 1 –6.

Oudeyer, P.Y. (2003). The production and recognition of emotions in speech: features and algorithms. *International Journal of Human-Computer Studies*, **59**, 157 – 183, ¡ce:title¿Applications of Affective Computing in Human-Computer Interaction¡/ce:title¿.

REFERENCES

Paiva, A. & Self, J. (1994). Tagus - a user and learner modeling workbench. *User Modeling and User-Adapted Interaction*, **4**, 197–226.

Pal, P., Iyer, A. & Yantorno, R. (2006). Emotion detection from infant facial expressions and cries. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 2, II.

Paliouras, G., Karkaletsis, V., Papatheodorou, C. & Spyropoulos, C.D. (1999). Exploiting learning techniques for the acquisition of user stereotypes and communities. In *UM99 User Modeling: Proceedings of the Seventh International Conference*, 169–178, Springer-Verlag.

Pallotta, V. (2004). Cognitive dialogue management. In *Natural Language Understanding and Cognitive Science*.

Panksepp, J. (1982). Toward a general psychobiological theory of emotions. *Behavioral and Brain Sciences*, **5**, 407–422.

Partala, T. & Surakka, V. (2004). The Effects of Affective Interventions in Human-Computer Interaction. *Interacting with Computers*, **16**, 295 – 309.

Perrault, C.R., Allen, J.F. & Cohen, P.R. (1978). Speech acts as a basis for understanding dialogue coherence. In *Proceedings of the 1978 workshop on Theoretical issues in natural language processing*, TINLAP '78, 125–132, Association for Computational Linguistics, Stroudsburg, PA, USA.

Petridis, S. & Pantic, M. (2008). Audiovisual discrimination between laughter and speech. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, 5117 –5120.

Petrushin, V. (2002). Creating emotion recognition agents for speech signal. In K. Dautenhahn, A. Bond, L. Canamero & B. Edmonds, eds., *Socially Intelligent Agents*, vol. 3 of *Multiagent Systems, Artificial Societies, and Simulated Organizations*, 77–84, Springer US.

Picard, R., Vyzas, E. & Healey, J. (2001). Toward machine emotional intelligence: analysis of affective physiological state. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **23**, 1175 –1191.

Picard, R.W. (2000). Toward computers that recognize and respond to user emotion. *IBM Systems Journal*, **39**, 705 –719.

Pittermann, J., Minker, W. & Pittermann, A. (2009). *Handling Emotions in Human-Computer Dialogues*. Springer.

Plutchik, R. (1960). The multifactor-analytic theory of emotion. *The Journal of Psychology*, **50**, 153–171.

Pohl, W. (1999). Logic-based representation and reasoning for user modeling shellsystems. *User Modeling and User-Adapted Interaction*, **9**, 217–282.

Polzin, T.S. & Waibel, A. (2000). Emotion-sensitive human-computer interfaces. In *ITRW on Speech and Emotion*.

Polzin, T.S. & Waibel, A.H. (1998). Detecting Emotions in Speech. In *International Conference on Cooperative Multimodal Communication*.

Pon-Barry, H. & Shieber, S. (2009). Identifying uncertain words within an utterance via prosodic features. In *Tenth Annual Conference of the International Speech Communication Association*.

Quinlan, J.R. (1986). Induction of decision trees. *Machine Learning*, **1**, 81–106, 10.1007/BF00116251.

Rabie, A., Wrede, B., Vogt, T. & Hanheide, M. (2009). Evaluation and discussion of multi-modal emotion recognition. In *Computer and Electrical Engineering, 2009. ICCEE '09. Second International Conference on*, vol. 1, 598 –602.

Rebolledo-Mendez, G., du Boulay, B. & Luckin, R. (2006). Motivating the learner: An empirical evaluation. *Lecture Notes in Computer Science*, **4053**, 545–554.

Reeves, B. & Nass, C. (1996). *How people treat computers, television, and new media like real people and places*. CSLI Publications and Cambridge university press.

Rich, E. (1979a). Building and exploiting user models. In *Proceedings of the 6th international joint conference on Artificial intelligence - Volume 2*, IJCAI'79, 720–722, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Rich, E. (1979b). User modeling via stereotypes. *Cognitive Science*, **3**, 329 – 354.

Rudnicky, A. & Xu, W. (1999). An agenda-based dialog management architecture for spoken language systems. In *IEEE Automatic Speech Recognition and Understanding Workshop*.

Russell, J. (1980). A circumplex model of affect. *Journal of personality and social psychology*, **39**, 1161.

Sarikaya, R. & Gowdy, J. (1998). Subband based classification of speech under stress. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, vol. 1, 569 –572 vol.1.

Satoh, N., Yamauchi, K., Matsunaga, S., Yamashita, M., Nakagawa, R. & Shinohara, K. (2007). Emotion clustering using the results of subjective opinion tests for emotion recognition in infants' cries. In *Proceedings of Interspeech*.

Schafer, R. (1968). *Aspects of Internalization*. International Universities Press.

Scherer, K.R. (1982). Emotion as a process: Function, origin and regulation. *Social Science Information*, **21**, 555–570.

Scherer, K.R. (2000). Psychological models of emotion. In *The Neuropsychology of Emotion*, Oxford University Press.

Schmalt, H., Sokolowski, K. & Langens, T. (2000). *Das Multi-Motiv-Gitter zur Erfassung von Anschluss, Leistung und Macht-MMG [The Multi-Motive-Grid of Connection, Performance and Power]*. Swets.

Schmitt, A., Pieraccini, R. & Polzehl, T. (2010a). "For Heaven's Sake, Gimme a Live Person!" Designing Emotion-Detection Customer Care Voice Applications in Automated Call Centers. In A. Neustein, ed., *Advances in Speech Recognition*, 191–219, Springer US, 10.1007/978-1-4419-5951-5_9.

Schmitt, A., Scholz, M., Minker, W., Liscombe, J. & Sündermann, D. (2010b). Is it Possible to Predict Task Completion in Automated Troubleshooters? In *Proc. of the International Conference on Speech and Language Processing (ICSLP)*.

Schneider-Düker, M. & Kohler, A. (1988). Die Erfassung von Geschlechtsrollen. Ergebnisse zur deutschen Neukonstruktion des Bern Sex-Role Inventory [Assessment of Sex Roles: Results of the German Version of the Bern Sex-Role Inventory]. *Diagnostica*, **34**, 256–270.

Schreck, J. (2003). *Security and Privacy in User Modeling*. Springer.

Schröder, M., Cowie, R., Heylen, D., Pantic, M., Pelachaud, C. & Schuller, B. (2008). Towards responsive sensitive artificial listeners. In *Fourth International Workshop on Human-Computer Conversation*, University of Sheffield, Sheffield, UK.

Schuller, B. (2006). *Automatische Emotionserkennung aus sprachlicher und manueller Interaktion*. Ph.D. thesis, TUM (Munich University of Technology).

Schuller, B. & Rigoll, G. (2006). Timing levels in segment-based speech emotion recognition. *Proc. Interspeech*, 1818–1821.

Schuller, B. & Rigoll, G. (2009). Recognising interest in conversational speech–comparing bag of frames and supra-segmental features. *Proc. Interspeech, Brighton, UK*, 1999–2002.

Schuller, B., Rigoll, G. & Lang, M. (2003). Hidden markov model-based speech emotion recognition. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, vol. 2, II – 1–4 vol.2.

Schuller, B., Müller, R., Lang, M. & Rigoll, G. (2005). Speaker independent emotion recognition by early fusion of acoustic and linguistic features within ensembles. *Proc. Interspeech*, 805–808.

Schuller, B., Müeller, R., Höernler, B., Höethker, A., Konosu, H. & Rigoll, G. (2007a). Audiovisual recognition of spontaneous interest within conversations. In *Proceedings of the 9th international conference on Multimodal interfaces*, ICMI '07, 30–37, ACM, New York, NY, USA.

Schuller, B., Rigoll, G., Grimm, M., Kroschel, K., Moosmayr, T. & Ruske, G. (2007b). Effects of in-car noise-conditions on the recognition of emotion within speech. *FORTSCHRITTE DER AKUSTIK*, **33**, 305.

Schuller, B., Seppi, D., Batliner, A., Maier, A. & Steidl, S. (2007c). Towards more reality in the recognition of emotional speech. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4, IV–941 –IV–944.

Sebe, N., Cohen, I., Gevers, T. & Huang, T. (2006). Emotion recognition based on joint visual and audio cues. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 1, 1136 –1139.

Seppänen, T., Väyrynen, E. & Toivanen, J. (2003). Prosody-based classification of emotions in spoken finnish. In *European Conference on Speech Communication and Technology*.

Sethu, V., Ambikairajah, E. & Epps, J. (2009). Pitch contour parameterisation based on linear stylisation for emotion recognition. In *Proceeedings of the 10th Annual Conference of the International Speech Communication Association (Interspeech)*.

Shortliffe, E. (1976). *Computer-based medical consultations: MYCIN*, vol. 388. Elsevier New York.

Sidorova, J. (2009). Speech emotion recognition with tgi+.2 classifier. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, EACL '09, 54–60, Association for Computational Linguistics, Stroudsburg, PA, USA.

Sleeman, D. (1985). Umfe: A user modelling front-end subsystem. *International Journal of Man-Machine Studies*, **23**, 71 – 88.

Song, M., Bu, J., Chen, C. & Li, N. (2004). Audio-visual based emotion recognition - a new approach. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, II–1020 – II–1025 Vol.2.

Steidl, S., Batliner, A., Schuller, B. & Seppi, D. (2009). The hinterland of emotions: Facing the open-microphone challenge. In *Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on*, 1 –8.

Steinberg, D., Budinsky, F., Paternostro, M. & Merks, E. (2008). *EMF: Eclipse Modeling Framework, 2nd Edition*. Addison-Wesley Professional.

Strobel, A., Beauducel, A., Debener, S. & Brocke, B. (2001). A German version of Carver and White's BIS/BAS scales. *Zeitschrift für Differentielle und Diagnostische Psychologie*, **22**, 216–227.

Takahashi, K. (2004). Remarks on svm-based emotion recognition from multi-modal bio-potential signals. In *Robot and Human Interactive Communication, 2004. ROMAN 2004. 13th IEEE International Workshop on*, 95 – 100.

Tan, J.W., Walter, S., Scheck, A., Hrabal, D., Hoffmann, H., Kessler, H. & Traue, H. (2011). Facial electromyography (femg) activities in response to affective visual stimulation. In *Affective Computational Intelligence (WACI), 2011 IEEE Workshop on*, 1 –5.

Tato, R., Santos, R., Kompe, R. & Pardo, J.M. (2002). Emotional space improves emotion recognition. In *In Proc. ICSLP 2002*, 2029–2032.

Tischler, M.A., Peter, C., Wimmer, M. & Voskamp, J. (2007). Application of emotion recognition methods in automotive research. In *Proceedings of the 2nd Workshop on Emotion and Computing – Current Research and Future Impact*.

Tomkins, S.S. (1963a). *Affect Imagery Consciousness: Volume I: The Positive Affects.*. Springer.

Tomkins, S.S. (1963b). *Affect Imagery Consciousness: Volume II: The Negative Affects.*. Springer.

Truong, K. & Van Leeuwen, D. (2007). An "open-set" detection evaluation methodology for automatic emotion recognition in speech. *Workshop on Paralinguistic Speech-between models and data*, 5–10.

Turunen, M. (2004). *Jaspis - A Spoken Dialogue Architecture and its Applications*. Ph.D. thesis, University of Tampere.

REFERENCES

Van Der Vlist, E. (2004). *Relax NG*. O'Reilly Media.

Van Melle, W.J. (1981). *System aids in constructing consultation programs*. UMI Research Press.

Ververidis, D. & Kotropoulos, C. (2004). Automatic speech classification to five emotional states based on gender information. *Proc. Eusipco, Vienna*, 341–344.

Vlasenko, B., Schuller, B., Mengistu, K., Rigoll, G. & Wendemuth, A. (2008). Balancing spoken content adaptation and unit length in the recognition of emotion and interest. *INTERSPEECH 2008 incorporating SST 2008, September 23*, **26**, 805–808.

Vogt, T. (2010). *Real-time Automatic Emotion Recognition from Speech*. Ph.D. thesis, Bielefeld University.

Vogt, T. & André, E. (2005). Comparing feature sets for acted and spontaneous speech in view of automatic emotion recognition. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, 474 –477.

Vogt, T., André, E. & Bee, N. (2008). Emovoice a framework for online recognition of emotions from voice. In E. André, L. Dybkjaer, W. Minker, H. Neumann, R. Pieraccini & M. Weber, eds., *Perception in Multimodal Dialogue Systems*, vol. 5078 of *Lecture Notes in Computer Science*, 188–199, Springer Berlin Heidelberg.

Wagner, J., Vogt, T. & André, E. (2007). A systematic comparison of different hmm designs for emotion recognition from acted and spontaneous speech. In A. Paiva, R. Prada & R. Picard, eds., *Affective Computing and Intelligent Interaction*, vol. 4738 of *Lecture Notes in Computer Science*, 114–125, Springer Berlin Heidelberg.

Wagner, J., Andre, E. & Jung, F. (2009). Smart sensor integration: A framework for multimodal emotion recognition in real-time. In *Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on*, 1 –8.

Walker, M.A., Cahn, J.E. & Whittaker, S.J. (1997). Improvising linguistic style: social and affective bases for agent personality. In *Proceedings of the first international conference on Autonomous agents*, AGENTS '97, 96–105, ACM, New York, NY, USA.

Walter, S., Hrabal, D., Scheck, A., Kessler, H., Bertrand, G., Nothdurft, F., Minker, W. & Traue, H. (2010). Individual emotional profiles in wizard-of-oz-experiments. In *Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '10, 68:1–68:2, ACM, New York, NY, USA.

Walter, S., Scherer, S., Schels, M., Glodek, M., Hrabal, D., Schmidt, M., Böck, R., Limbrecht, K., Traue, H. & Schwenker, F. (2011). Multimodal emotion classification in naturalistic user behavior. In J. Jacko, ed., *Human-Computer Interaction. Towards Mobile and Intelligent Interaction Environments*, vol. 6763 of *Lecture Notes in Computer Science*, 603–611, Springer Berlin Heidelberg.

Wang, H.L. & Cheong, L.F. (2006). Affective understanding in film. *Circuits and Systems for Video Technology, IEEE Transactions on*, **16**, 689 – 704.

Wang, Y. & Guan, L. (2005). Recognizing human emotion from audiovisual information. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, vol. 2, ii/1125 – ii/1128 Vol. 2.

Watson, D., Clark, L.A. & Tellegen, A. (1988). Development and validation of brief measures of positive and negative affect: The panas scales. *Journal of Personality and Social Psychology*, **54**, 1063–1070.

Wöllmer, M., Eyben, F., Reiter, S., Schuller, B., Cox, C., Douglas-Cowie, E. & Cowie, R. (2008). Abandoning emotion classes-towards continuous emotion recognition with modelling of long-range dependencies. In *Ninth Annual Conference of the International Speech Communication Association*.

Wrede, B. & Shriberg, E. (2003). Spotting "hot spots" in meetings: Human judgments and prosodic cues. In *in Proc. Eurospeech*, 2805–2808.

Wu, S., Falk, T. & Chan, W. (2008). Long-term spectro-temporal information for improved automatic speech emotion classification. In *Ninth Annual Conference of the International Speech Communication Association*.

Wu, W., Zheng, T.F., Xu, M.X. & Bao, H.J. (2006). Study on speaker verification on emotional speech. In *Proceedings of Interspeech*.

Wundt, W.M. (1902). *Grundzüge der physiologischen Psychologie*. Engelmann.

Xiao, Z., Dellandrea, E., Dou, W. & Chen, L. (2010). Multi-stage classification of emotional speech motivated by a dimensional emotion model. *Multimedia Tools and Applications*, **46**, 119–145.

Yildirim, S., Narayanan, S. & Potamianos, A. (2011). Detecting emotional state of a child in a conversational computer game. *Computer Speech & Language*, **25**, 29 – 44, ¡ce:title¿Affective Speech in Real-Life Interactions¡/ce:title¿.

You, M., Chen, C., Bu, J., Liu, J. & Tao, J. (2007). Manifolds based emotion recognition in speech. *Computational Linguistics and Chinese Language Processing*, **12**, 49–64.

Yun, S. & Yoo, C. (2009). Speech emotion recognition via a max-margin framework incorporating a loss function based on the watson and tellegen's emotion model. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, 4169 –4172.

Zeng, Z., Hu, Y., Roisman, G., Wen, Z., Fu, Y. & Huang, T. (2007). Audio-visual spontaneous emotion recognition. In T. Huang, A. Nijholt, M. Pantic & A. Pentland, eds., *Artifical Intelligence for Human Computing*, vol. 4451 of *Lecture Notes in Computer Science*, 72–90, Springer Berlin Heidelberg.

Zeng, Z., Pantic, M., Roisman, G. & Huang, T. (2009). A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **31**, 39 –58.

Zhang, T., Hasegawa-Johnson, M. & Levinson, S.E. (2003). Mental state detection of dialogue system users via spoken language. In *ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*.

Zhou, G., Hansen, J. & Kaiser, J. (2001). Nonlinear feature based classification of speech under stress. *Speech and Audio Processing, IEEE Transactions on*, **9**, 201 –216.

Zhou, J., Wang, G., Yang, Y. & Chen, P. (2006). Speech emotion recognition based on rough set and svm. In *Cognitive Informatics, 2006. ICCI 2006. 5th IEEE International Conference on*, vol. 1, 53 –61.

Zuckerman, M., Porac, J., Lathin, D., Smith, R. & L., D.E. (1978). On the importance of self-determination for intrinsically-motivated behavior. *Personality and Social Psychology Bulletin*, **4**, 443–446.

Declaration

I herewith declare that I have produced this paper without the prohibited assistance of third parties and without making use of aids other than those specified; notions taken over directly or indirectly from other sources have been identified as such. This paper has not previously been presented in identical or similar form to any other German or foreign examination board.

The thesis work was conducted from February 2009 to May 2013 under the supervision of Prof. Dr. Dr.-Ing. Wolfgang Minker at Ulm University.

Ulm,