

Context-based Multimodal Interpretation: An Integrated Approach to Multimodal Fusion and Discourse Processing

Dissertation
zur Erlangung des Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
der Naturwissenschaftlich-Technischen Fakultäten der Universität des Saarlandes

vorgelegt von

Norbert Pflieger

Saarbrücken 2007

Datum des Kolloquiums: 27.11.2007

Dekan und Vorsitzender: Prof. Dr. Thorsten Herfet

Gutachter:

1. Prof. Dr. Dr. h.c. mult. Wolfgang Wahlster

2. Prof. Dr. Manfred Pinkal

Akademischer Beisitzer: Dr. Dominik Heckmann

Eidesstattliche Erklärung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.

Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form in einem Verfahren zur Erlangung eines akademischen Grades vorgelegt.

Saarbrücken, 27. November 2007

Norbert Pflieger

Acknowledgements

First of all, I want to thank Professor Wahlster for giving me the opportunity to write this thesis in the stimulating research environment here at the DFKI. I am particularly thankful for the interesting discussions and for the valuable insights you provided. I also want to thank Professor Pinkal for taking over the role of second corrector of this thesis.

I would like to thank my supervisors at DFKI Jan Alexandersson and Norbert Reithinger for their support and encouragement during the last years and for helping me with their expertise. Jan, you are the one who brought me to the field of multimodal dialogue systems in the first place and so it was you who laid the foundation for this thesis. Special thanks also go to my friends and colleagues Alexander Pfalzgraf, Jan Schehl and Jochen Steigner. You were the ones that dealt with versions of the manuscript in different stages and I am thankful for all the time and effort you not only put into proof-reading this thesis but also into supporting me with your ideas and never ending-patience and more than a few rounds of playing football or poker to take off some of the pressure while writing this thesis.

I also want to thank Markus Löckelt, Elsa Pecourt and Massimo Romanelli for their friendship, collaboration and support during the last years. Additional thanks go to Tilman Becker, Anselm Blocher, Ralf Engel, Gerd Herzog, Alassane Ndiaye and all the other colleagues at the Intelligent User Interface lab at DFKI. There also have been many colleagues involved in the different projects and to all of them I am thankful for discussions and their professional support.

I also want to thank Kate Flynn who, as a native speaker, went looking for typos and “Denglisch” and thereby gave the last polishing to the manuscript. I want to thank my friends and former flatmates who supported me over all the years. Special thanks go to my family: to my parents, who always took such an unfailing interest in my work; to my sister and my brother. You all stood by me during this busy time. Thank you! But most of all I want to thank my wife Alex. Without your patience and support, I would never have gotten through all of this.

Kurzzusammenfassung

Diese Dissertation befasst sich mit der kontextbasierten Interpretation von verbalen und non-verbalen Gesprächsbeiträgen im Rahmen von multimodalen Dialogsystemen. Im Rahmen dieser Arbeit wird, basierend auf einer detaillierten Analyse multimodaler Diskursphänomene, ein umfassendes Modell des Gesprächskontextes erarbeitet. Dieses Modell soll sowohl die Verarbeitung einer Vielzahl von referentiellen und elliptischen Ausdrücken, als auch die Erzeugung reaktiver Aktionen wie sie für den Sprecherwechsel benötigt werden unterstützen. Ein zentrales Ziel dieser Arbeit ist die Entwicklung einer generischen Komponente zur multimodalen Fusion und Diskursverarbeitung. Anhand der Integration dieser Komponente in drei unterschiedliche Dialogsysteme soll der generische Charakter dieser Komponente gezeigt werden.

Short Abstract

This thesis is concerned with the context-based interpretation of verbal and nonverbal contributions to interactions in multimodal multiparty dialogue systems. On the basis of a detailed analysis of context-dependent multimodal discourse phenomena, a comprehensive context model is developed. This context model supports the resolution of a variety of referring and elliptical expressions as well as the processing and reactive generation of turn-taking signals and the identification of the intended addressee(s) of a contribution. A major goal of this thesis is the development of a generic component for multimodal fusion and discourse processing. Based on the integration of this component into three distinct multimodal dialogue systems, the generic applicability of the approach is shown.

Zusammenfassung

Diese Dissertation beschäftigt sich damit, ein umfassendes Kontextmodell zu entwickeln, welches die Interpretation und Integration von verbalen und nonverbalen Beiträgen im Rahmen von multimodalen Dialogsystemen ermöglicht. Wenn Menschen sich miteinander unterhalten nutzen sie eine Vielzahl an nonverbalem Verhalten wie beispielsweise wechselnde Gesichtsausdrücke, Augenkontakt, Kopf- oder Handbewegungen, die sowohl zum propositionalen, als auch interaktionalen Inhalt einer Äußerung beitragen. Da viele dieser Handlungen unterschiedliche Bedeutungen haben können, je nachdem, in welcher Situation sie gebraucht werden, müssen sie jeweils im Kontext ihrer Verwendung interpretiert werden. Neben der kontextbasierten Interpretation von nonverbalen Handlungen verlangen einige Handlungen eine unmittelbare Reaktion vom Zuhörer. Die Interpretation von verbalen Aspekten einer Äußerung erfordern darüber hinaus Kenntnis über den vorangegangenen Diskurs, um referentielle und elliptische Äußerungen auflösen zu können.

Ein wesentliches Ziel dieser Dissertation ist die Realisierung einer generischen Komponente, die in der Lage ist, diese kontextbasierte Interpretation von Äußerungen während eines Diskurses vorzunehmen und die außerdem reaktives Feedback auslösen kann. In Rahmen dieser Arbeit werden mehrere wissenschaftliche Zielsetzungen und Fragestellungen angegangen. Zunächst einmal beschäftigen wir uns mit der Frage, was die relevanten verbalen und nonverbalen Handlungen sind, die zum Kontext von multimodalen Dialogsystemen beitragen, an denen mehrere Personen beteiligt sind. Indem wir die Schlüsselcharakteristika von verbalen und nonverbalen Handlungen betrachten, können wir eine Basis schaffen für eine integrierte Verarbeitung der gesamten Kommunikationssituation. Die zweite Fragestellung, die hier angegangen wird, beschäftigt sich mit den Faktoren, die den Kontext von Mensch-Maschine Interaktionen definieren. Im Besonderen werden wir uns damit beschäftigen, bis zu welchem Grad die physische Umgebung, physische Handlungen und das ontologische Weltwissen in das Kontextmodell miteinbezogen werden müssen. Das daraus resultierende Kontextmodell schließt zwei Typen von aktivationsbasierten Kontextrepräsentationen ein: den unmittelbaren Gesprächskontext sowie den allgemeinen Gesprächsverlauf. Der unmittelbare Gesprächskontext repräsentiert hierbei den momentanen physikalischen und sprachspezifischen Kontext, während der Gesprächsverlauf die vorhergehenden Beiträge der einzelnen Gesprächsteilnehmer repräsentiert. Neben diesen zwei Arten von Kontext beinhaltet das Kontextmodell außerdem ein so genanntes Langzeitgedächtnis. Dieses Gedächtnis umfasst das Wissen des Dialogsystems in einem semantischen Netzwerk. Wenn ein Eintrag aus dem Langzeitgedächtnis während eines Gesprächs erwähnt wird, können damit assoziierte Objekte aktiviert werden.

Basierend auf dem Kontextmodell stellen wir eine integrierte multimodale Fusions- und Diskurskomponente (FADE) vor, die in der Lage ist, eine Vielzahl von kontextabhängigen Gesprächsphänomenen zu behandeln wie sie in natürlichen multimodalen Dialogen vorkom-

men, an denen mehrere Personen beteiligt sind. Die funktionale Architektur von Fade besteht aus zwei Verarbeitungsschichten: (i) das Produktionsregelsystem PATE, das für die reaktive Interpretation von wahrgenommenen monomodalen Ereignissen verantwortlich ist, und (ii) einem Diskursgedächtnis (DiM), das nicht nur dafür verantwortlich ist, eine kohärente Repräsentation des gerade stattfindenden Gesprächs aufrechtzuerhalten, sondern auch für die Fusion von multimodalen Beiträgen und für die Auflösung von referentiellen und elliptischen Äußerungen zuständig ist. Das Produktionsregelsystem PATE wurde ursprünglich entwickelt, um den reaktiven Teil von FADE und die generelle Ablaufsteuerung zu realisieren. PATE kann jedoch auch zur Verwirklichung einer Vielzahl von Komponenten in multimodalen Dialogsystemen verwendet werden. Dies wurde zum Beispiel mit dem SAMMIE-System unter Beweis gestellt, das innerhalb des TALK-Projekts entwickelt worden ist. Darin wurden alle funktionalen Schlüsselkomponenten mit Hilfe von PATE implementiert.

Ein entscheidender Gesichtspunkt des hier vorgestellten Algorithmus zur Auflösung referentieller Ausdrücke ist die Integration von zwei normalerweise getrennten Aufgaben, genauer gesagt von multimodaler Fusion und Referenzauflösung. Typischerweise werden diese beiden Aufgaben in einem multimodalen Dialogsystem nacheinander ausgeführt. Zuerst werden die monomodalen Beiträge durch eine kontextunabhängige Fusionskomponente integriert, bevor die noch verbliebenen referentiellen Ausdrücke durch eine Diskursverarbeitungs-komponente aufgelöst werden. Diese Fragmentierung in zwei Komponenten ist jedoch künstlich und entspricht nicht der Art und Weise wie Menschen normalerweise verbale und nonverbale Beiträge in einer Äußerung verwenden. Im Grunde genommen realisiert eine solche Fusionskomponente eine spezialisierte Referenzauflösung für deiktische Ausdrücke. Die Tatsache jedoch, dass es deiktische Ausdrücke gibt, die entweder durch eine Zeigegeste oder durch eine Referenz, die während des vorangegangenen Diskurses eingeführt worden ist, aufgelöst werden können, unterstreicht die Notwendigkeit eines integrierten Ansatzes zur Referenzauflösung. Aus diesem Grund präsentieren wir einen multimodalen Referenzauflösungsalgorithmus, der in der Lage ist, eine Vielzahl von referentiellen Ausdrücken wie zum Beispiel deiktischen, exophorischen, anaphorischen, räumlichen und zeitlichen Referenzen zu verarbeiten.

Ein weiteres Ziel dieser Arbeit ist es, die Lücke zwischen kommerziellen und wissenschaftlichen Dialogsystemen zu überbrücken. Das bedeutet, dass die hier vorgestellten Arbeiten geleitet werden durch die Idee, einen generischen Baukasten zu realisieren, der einfach an neue Systeme und Anwendungen angeschlossen werden kann. Effizienz, Robustheit, Flexibilität und schnelle Reaktionszeiten werden dabei als die zentralen Anforderungen an den Entwicklungsprozess angesehen. Auf diese Weise berührt diese Arbeit drei Felder im Bereich der multimodalen Dialogsysteme: (i) multimodale Fusion, (ii) Kontextmodellierung und multimodale Diskursverarbeitung und (iii) die Überbrückung der Kluft zwischen kommerziellen und wissenschaftlichen Dialogsystemen.

Die Ergebnisse dieser Dissertation sind in drei verschiedene multimodale Dialogsysteme integriert worden: VirtualHuman, SmartWeb und OMDIP. Der Fokus von VirtualHuman liegt auf Interaktionen, an denen mehrere Personen beteiligt sind, da das System es erlaubt, dass bis zu zwei menschliche Benutzer mit drei virtuellen Charakteren interagieren. SmartWeb ist ein dyadisches Dialogsystem, das einen intelligenten mobilen Informationsservice für verschiedene Domänen verwirklicht. OMDIP ist ebenfalls ein dyadisches Dialogsystem, das mobilen Zugang zu webbasierten Domänen ermöglicht. Durch die Integration von FADE in diese drei so verschiedenen Dialogsysteme konnte der generische Charakter von FADE aufgezeigt werden. Die Ergebnisse dieser Forschung wurden in 22 Artikeln veröffentlicht.

Abstract

This dissertation addresses the problem of designing a comprehensive context model that enables the interpretation and integration of verbal and nonverbal contributions during dyadic as well as multiparty human-computer interactions. When people engage in conversation, they make use of a variety of nonverbal behavior (e.g., facial expressions, eye gaze, head movement, hand gestures, etc.) that contribute both to the propositional and interactional content of an utterance. Since many of these actions can have different meanings depending on the situation, they have to be interpreted within their context of use. Besides the context-based interpretation of nonverbal actions, some of them also require the listener to show reactive behavior. Moreover, the verbal aspects of an utterance also require access to the previous discourse context in order to be able to resolve referring or elliptical expressions. The ultimate goal of this dissertation is the realization of a generic component that is able to perform the context-based interpretation of contributions to a discourse and that is also able to trigger reactive feedback if appropriate.

In this thesis a number of research objectives and questions are addressed. First of all, we are concerned with the question of what the relevant verbal and nonverbal actions that contribute to the context of multimodal multiparty dialogue systems are. By considering the key characteristics of verbal and nonverbal actions, we lay the basis for an integrated processing of the total communicative situation. The second research question addresses the factors that define the context of human-computer interaction. We will particularly investigate to what extent the physical environment, physical actions and ontological world knowledge have to be incorporated into the context model. Another objective is the realization of an architecture that supports both the interpretation of verbal and nonverbal contributions and the triggering of reactive behavior (e. g., turn-taking and backchannel feedback). Moreover, we focus on the development of a generic algorithm for resolving a wide variety of referring expressions.

The resulting context model comprises two types of activation-based context representations: the immediate conversational context and the discourse context. The immediate conversational context represents the current physical, perceptual and conversational context and the discourse context represents the previous contributions of the individual participants. Each object has an activation value associated with it that determines its current accessibility and the entries in both context representations are organized by means of their activation values. Besides the two types of context, the context model also incorporates a so-called long term memory. This memory encompasses all knowledge a dialogue system is aware of in a semantic network. If an entry of the long term memory is mentioned during a discourse, associated objects can also be activated by using activation spreading.

Based on the context model, we introduce an integrated multimodal fusion and discourse processing engine (FADE) that is capable of processing a wide range of context-dependent conversational phenomena as they occur in natural multimodal multiparty dialogues. The

high-level functional architecture of FADE consists of two processing layers: (i) the production rule system PATE which is responsible for the reactive interpretation of perceived monomodal events and (ii) a discourse modeler (called DiM) that is responsible for maintaining a coherent representation of the ongoing discourse, for the fusion of multimodal contributions and for the resolution of referring and elliptical expressions.

The production rule system PATE has originally been designed and implemented to deal with the reactive part of FADE and to realize the control-flow. However, PATE can also be used to realize a wide variety of components of multimodal dialogue systems. This has been demonstrated with the SAMMIE system that was developed within the TALK project. Here, all functional key components are implemented by means of PATE.

A key aspect of the reference resolution algorithm that is presented in this thesis is the integration of two usually separated tasks, i. e., multimodal fusion and reference resolution. Typically, these two tasks are performed in a sequential fashion. First, monomodal contributions are fused by a context-independent fusion component before the remaining referring expressions are resolved by a discourse processing component. This fragmentation into two components, however, is artificial and does not reflect the way humans use verbal and non-verbal contributions in utterances. A multimodal fusion component actually performs a specialized reference resolution as it resolves deictic expressions by using nonverbal actions. The fact that there are deictic expressions that can either be resolved by a pointing gesture or by a referent that was introduced during the previous discourse underlines the need for an integrated processing of referring expressions. To this end, we present a multimodal reference resolution algorithm that is able to process a variety of referring expressions like deictic, exophoric, anaphoric, spatial and temporal references by incorporating the aforementioned comprehensive context model.

A further goal of this thesis is to contribute to bridging the gap between commercial and research dialogue systems. This means that the work presented here is guided by the idea of developing a generic toolkit that is easy to adapt to new systems or applications. Efficiency, robustness, flexibility and promptness of reaction are hereby considered to be the key requirements for the developing process. Thus, this thesis touches three fields in the area of multimodal dialogue systems: (i) multimodal fusion, (ii) context modeling and multimodal discourse processing, and (iii) bridging the gap between commercial and research dialogue systems.

The results of this dissertation have been integrated into three different multimodal dialogue systems: VirtualHuman, SmartWeb, and OMDIP. The focus of VirtualHuman is on multiparty discourse as it allows up to two human users to interact with three virtual characters. SmartWeb is a dyadic dialogue system that realizes an intelligent mobile information service for various domains. OMDIP is a dyadic dialogue system that provides mobile access to web-based applications. Based on the integration of FADE into these three diverse dialogue systems we underpin the generic toolkit character of FADE. The results of this research have been published in 22 articles and papers.

Contents

1	Introduction	1
1.1	Multimodal Fusion	1
1.2	Context-based Interpretation of Multimodal Utterances	2
1.3	Bridging the Gap between Commercial and Research Dialogue Systems	3
1.4	System Context	3
1.5	Main Objectives	7
1.6	Thesis Outline	8
2	Verbal and Nonverbal Behavior	11
2.1	Introduction	11
2.1.1	Communicative Functions vs. Communicative Behavior	12
2.1.2	Interactional vs. Propositional Content	12
2.1.3	Deliberative vs. Reactive Behavior	12
2.1.4	Meaning and Context	13
2.2	Characteristics of Verbal Contributions	14
2.2.1	Conveying Meaning—The Propositional Content of an Utterance	14
2.2.2	Conversational Implicature	15
2.2.3	On Speech Act Theory and Propositions	15
2.2.4	Dialogue Acts	16
2.2.5	The Context Changing Function of Verbal Contributions	17
2.3	Characteristics of Nonverbal Contributions	17
2.3.1	Classifying Nonverbal Behavior	18
2.3.2	Gestures	18
2.3.3	Eye Behavior	23
2.3.4	Facial Expressions	23
2.3.5	Vocal Behavior	23
2.3.6	Body Posture	24
2.3.7	Physical Actions	25
2.3.8	Functions of Nonverbal Behavior	25
2.3.9	The Context Update Function of Nonverbal Behavior	27
2.4	The Total Communicative Process	27
2.4.1	Interrelation of Verbal and Nonverbal Behavior	27
2.4.2	Mutual Disambiguation	28
2.4.3	The Context Changing Function of Utterances	28
2.5	Conclusion and Discussion	29
2.5.1	Our Notion of Verbal and Nonverbal Behavior	29

2.5.2	Developing the Argument for an Integrated Multimodal Fusion and Discourse Processing Component	29
3	Characteristics of Discourse	31
3.1	Introduction	31
3.1.1	Types of Discourse	31
3.1.2	Social Roles and Status in Dialogues	35
3.2	Propositional Aspects of Discourse	36
3.2.1	Drawing Inferences	36
3.2.2	The Structure of Discourse	37
3.2.3	Referring Expressions	41
3.2.4	Ellipsis and Fragments	53
3.2.5	Common Ground	55
3.3	Interactional Aspects of Dialogue	61
3.3.1	Introduction	61
3.3.2	Turn-Taking	62
3.3.3	Identifying the Addressee(s)	66
3.3.4	Backchannel Feedback	68
3.4	Conclusion and Discussion	70
3.4.1	Developing the Argument for an Integrated Multimodal Fusion and Discourse Processing Component	71
4	Computational Approaches to Multimodal Fusion and Discourse Modeling	73
4.1	Introduction	73
4.1.1	Multimodal Dialogue Systems: Architectures and Information-flow	75
4.1.2	Designing Multimodal Interfaces—Human Factors	79
4.1.3	Mutual Disambiguation and Error Handling in Multimodal Interfaces	80
4.2	Context Modeling	81
4.2.1	Dimensions of Context	81
4.2.2	Computational Context Models	84
4.2.3	Summary	87
4.3	Multimodal Fusion	87
4.3.1	Introduction	88
4.3.2	Input Modalities	88
4.3.3	Computational Approaches to Multimodal Fusion	97
4.3.4	End-Of-Turn Detection	102
4.3.5	Interaction Patterns and Input Synchronization	103
4.3.6	Summary	104
4.4	Approaches to Discourse Modeling and Processing	104
4.4.1	Discourse Structure in Human-Computer Dialogues	105
4.4.2	Contextual Enrichment	105
4.4.3	Computational Approaches to Discourse Modeling	107
4.4.4	Summary	115
4.5	Multiparty Human-Computer Interactions	116
4.5.1	Computational Approaches to Turn-Taking Management	116
4.5.2	Addressee Identification in Dialogue Systems	117
4.6	Conclusion and Discussion	118

5	Representing Meaning in Multimodal Dialogue Systems	121
5.1	Introduction	121
5.2	Representation Notations, Languages and Formats	122
5.2.1	RDF	123
5.2.2	RDF Schema	123
5.2.3	Web Ontology Language—OWL	123
5.2.4	Typed Feature Structures	125
5.2.5	Discussion	133
5.3	Modeling Meta Information	134
5.4	Ontology-based Representation of Meaning	136
5.4.1	The SmartKom Ontology	136
5.4.2	The SmartWeb Ontology	138
5.5	Conclusion and Discussion	139
6	A Comprehensive Context Model for Multimodal Multiparty Discourse	141
6.1	Introduction	141
6.1.1	Taking the Perspective of the Participants	142
6.1.2	Immediate Conversational Context vs. Discourse Context	142
6.2	The Immediate Conversational Context	142
6.2.1	Representing Dialogue Participants	143
6.2.2	Modeling the Conversational Status	144
6.2.3	Modeling the Physical Environment	145
6.3	The Discourse Model	149
6.3.1	Modality Layer	149
6.3.2	Referential Layer	150
6.3.3	Discourse Objects	150
6.4	The Long-Term Memory	151
6.5	The Context Changing Function of Utterances, Physical Actions and Events	152
6.5.1	Updating the Immediate Conversational Context	153
6.5.2	Updating the Discourse Context	153
6.6	Summary	154
7	PATE—A Production Rule System Based on Typed Feature Structures	155
7.1	Introduction	155
7.1.1	Development of PATE	156
7.1.2	Extending Classic Production Rule Systems	156
7.1.3	Related Work	157
7.1.4	Requirement Analysis	157
7.2	The Architecture of PATE	158
7.2.1	The Working Memory	159
7.2.2	The Goal-Stack	159
7.2.3	The Long-Term Memory	160
7.3	Production Rules	162
7.3.1	Condition Part	163
7.3.2	Action Part	164
7.4	Applying Production Rules	166
7.4.1	Computing the Conflict Set	166

7.4.2	Selecting the Firing Rule	166
7.4.3	Executing the Actions of the Firing Rule	167
7.4.4	Updating the Working Memory through External Events	167
7.5	Supporting the Process of Rule Development	168
7.5.1	The Graphical User Interface	168
7.6	Data Representation in PATE	171
7.6.1	Requirement Analysis	171
7.6.2	Extended Typed Feature Structures	172
7.6.3	The Type System	172
7.6.4	Multiple Slot-Fillers	173
7.6.5	Unique Identifiers	173
7.6.6	Activation	173
7.6.7	Supported Operations on eTFS Objects	173
7.7	Using PATE as a Generic Component for Multimodal Dialogue Systems	176
7.7.1	SAMMIE—An In-Car Multimodal Dialogue System	176
7.8	Conclusion	177
8	An Integrated Fusion and Discourse Engine	179
8.1	Introduction	179
8.2	An Ontological Model of Conversational Dialogue	180
8.2.1	Modeling Verbal and Nonverbal Actions	181
8.2.2	Modeling the Conversational Status	182
8.2.3	Modeling the Physical Environment	182
8.2.4	Modeling Referring Expressions	183
8.2.5	Utility Classes	185
8.3	The Basic Architecture of FADE	185
8.4	The Perception Module	187
8.4.1	Determining the Activation of Verbal and Nonverbal Actions	187
8.4.2	Determining Visual Salience	187
8.4.3	Event Routing	190
8.5	The Immediate Conversational Context	190
8.5.1	Representation and Maintenance of the Conversational State	190
8.5.2	Identifying the Intended Addressee(s)	192
8.5.3	Robust and Generic End-Of-Turn Detection	193
8.6	The Discourse Context	194
8.6.1	A Layered Multimodal Discourse Model	195
8.6.2	Resolution of Referring Expressions	198
8.6.3	Resolving Elliptical Expressions and Fragments	209
8.6.4	Updating the Discourse Context	211
8.6.5	Retrieving the Referential and Grounding Status of Referential Objects	212
8.6.6	Disambiguation of Multiple Interpretation Hypotheses	212
8.6.7	Contextual Enrichment and Validation	214
8.7	The Long-Term Memory	214
8.7.1	Transferring Information from the LTM to the WM	214
8.7.2	Updating the LTM	215
8.8	Conclusion	215

9	Applying FADE in Multimodal Dialogue Systems	217
9.1	Introduction	217
9.1.1	Initializing the System	217
9.1.2	The Central Configuration File	218
9.1.3	Importing the Type System	218
9.1.4	Registering Communication Channels	218
9.1.5	Initializing the Immediate Conversational Context	219
9.2	FADE in the VirtualHuman System	219
9.2.1	The Scenario of VirtualHuman	220
9.2.2	Tasks of FADE in VirtualHuman	222
9.2.3	The Architecture of the VirtualHuman System	223
9.2.4	The CDE Framework	225
9.2.5	FADE's Processing Strategies for VirtualHuman	228
9.3	FADE in the SmartWeb System	231
9.3.1	The Architecture of the SmartWeb System	232
9.3.2	Tasks of FADE in SmartWeb	235
9.3.3	Processing Strategies of FADE in SmartWeb	236
9.4	FADE in the OMDIP System	241
9.4.1	The OMDIP System	241
9.4.2	The Architecture of the OMDIP System	243
9.4.3	FADE's Interfaces in OMDIP	245
9.4.4	The Processing Strategy of FADE in OMDIP	247
9.4.5	Step-by-Step Example of the Application of Rules	248
9.4.6	Changes in the Second Version of OMDIP	250
9.5	Summary	250
10	Conclusion and Discussion	253
10.1	Contributions	253
10.1.1	Main Scientific and Practical Results	254
10.1.2	Publications	256
10.2	Future Work	257
A	Trace of FADE in the SmartWeb System	259
B	FADE's Rule Base for OMDIP	281
11	Bibliography	285

List of Figures

1.1	Screen-shot of the first version of the VIRTUALHUMAN system in action. The teacher (on the right) teaches both the pupil (on the left) and the human user, who is in front of the system.	4
1.2	The left part of the figure shows the second version of the VIRTUALHUMAN system during the first game phase. The right part shows game phase two.	5
1.3	Sample interaction with the SMARTWEB system: The left part depicts the result of the user request “ <i>Who won the World Cup in 2002?</i> ”; the right part depicts the result of the follow-up question “ <i>How often did this team win the World Cup?</i> ”	6
1.4	The OMDIP system in action: The user selects a <i>Soundlogo</i> using the pen while uttering the request to play the selected Soundlogo.	7
2.1	Kendon’s continuum (adapted from McNeill (1992)).	19
2.2	Example of an emblematic gesture (from Kempe (2005)).	20
2.3	Example of a deictic pointing gesture (from Kempe (2005)).	21
2.4	Example of an iconic gesture. The speaker on the right indicates the height of an object (from Kempe (2005)).	21
2.5	Example of a metaphoric gesture (from Kempe (2005)).	22
2.6	A taxonomy of verbal and nonverbal actions.	29
2.7	Categorization of verbal and nonverbal contributions with respect to their contribution to the context of an interaction.	30
3.1	Types and subtypes of dialogue (adapted from Walton (1992)).	34
3.2	The conversational roles of participants in multiparty dialogues.	34
3.3	Example of a right frontier of a tree: The filled circles mark the right frontier of the tree.	38
4.1	Extended Architecture for a multimodal dialogue system (from Bunt et al. (2005)).	75
4.2	Abstract representation of the architecture and information-flow of a multimodal dialogue system.	76
4.3	Extended, reactive architecture for a multimodal dialogue system (Allen et al., 2001).	78
4.4	The three layers or <i>channels</i> of information transfer in multimodal communication (from Thórisson (1994)).	79
4.5	Example discourse state and segmented interaction for a COLLAGEN video cassette recorder agent (taken from Rich et al. (2001)).	85

4.6	Three examples of in-car controller knobs for controlling secondary vehicle systems. From left to right: The BMW iDrive, the Audi MMI and the Mercedes Command.	89
4.7	Example of a data glove: The 5DT Data Glove 16.	93
4.8	Example application of Immersion’s TouchSense screen.	94
4.9	The Phantom Omni haptic device.	94
4.10	Prototype of a haptic interface that is integrated into the handlebars of a motorbike. The user can rotate the wheel, push and pull it.	95
4.11	The tangible user interface COHIBIT. The virtual characters help the user to assemble a car on the workbench in the middle of the picture.	96
4.12	Sample interaction with the <i>Mobile ShopAssist</i> (MSA).	97
4.13	The architecture of the fusion component for the Mobile Shopping Assistant (taken from Wasinger et al. (2005)).	101
4.14	Temporal synchronization patterns of the Mobile Shopping Assistant (taken from Wasinger et al. (2005)).	102
4.15	The Three-Tiered Discourse Representation: Orientation of Discourse Pegs to other objects (adapted from Luperfoy (1991)).	108
4.16	Example of a mental representation in the approach of Salmon-Alt (2000). . .	109
4.17	The multimodal context representation of SmartKom: The dashed arrow indicates that the value of the broadcast in the (new) structure on the right is shared with that of the old one (on the left).	110
4.18	The MRE addressee identification algorithm (adapted from Traum (2004)). .	119
5.1	Example of an RDF document describing an instance of a football-player in a specific match (the ellipses mark omitted information).	123
5.2	Example RDFS excerpt defining the class <i>FieldMatchFootballPlayer</i> and one of its properties.	124
5.3	Feature structure based representation of a song.	125
5.4	Typed feature structure based representation of a song.	125
5.5	TFS exemplifying the co-reference mechanism of typed feature structures. . .	126
5.6	Example of the unification of two TFSs.	127
5.7	Example of a failed unification.	127
5.8	Example of a failed restricted unification.	128
5.9	Example of overlay where unification would fail.	129
5.10	Overlay of two TFSs that have conflicting types.	131
5.11	Top-level part of the SMARTKOM ontology.	137
5.12	Overview of SmartSUMO/SWIntO as a UML package diagram. Packages represent ontology modules; dotted lines represent dependencies between modules (taken from Oberle et al. (2006)).	138
5.13	LingInfo model with example domain ontology classes and LingInfo instances (simplified) (taken from Buitelaar et al. (2006)).	139
6.1	Graphical representation of the spatial relations between an object of the physical environment and its neighboring objects. The left part of the figure shows the coarse-grained model using six relations and the right part shows the fine-grained model using ten relations. Due to layout restrictions, both figures do not show the <i>above</i> relation.	147

6.2	Example configuration of a physical environment: The football player Ronaldo is either to the right of Roberto Carlos or to the left of him, depending on the point of view.	148
6.3	Basic organization of the context model (taken from (Pfleger and Alexander- sson, 2006)): The left part of the figure shows an excerpt of the long-term memory and the right part shows some objects on the Referential Layer in the working memory. Both memory structures are separated by an activation threshold.	152
7.1	Basic architecture of PATE. The system comprises three interweaved memory areas: (i) a working memory, (ii) a goal-stack and (iii) a long-term memory. The applicability of the production rules is tested in two steps: First, a goal conflict set is computed that encompasses all rules whose goal condition is fulfilled. Then, based on this preselection, the actual conflict set is determined. The highest ranked rule is selected to fire which possibly results in changes to the working memory and the goal-stack.	159
7.2	Example of Priority Relations for defining primary associations between an instance of a class and a subset of its slots (from Gehrmann (2007)).	162
7.3	The graphical user interface of PATE.	168
7.4	eTFS structure exemplifying the duplicate usage of the same identifier “artist42”.	174
7.5	System architecture of the SAMMIE system.	177
7.6	Sample type-hierarchy where overlay would return more than one result (adapted from Alexandersson and Becker (2003)). Since the types t_3 and t_4 have no unique least upper bound (LUB) but two minimal upper bounds (MUBs), the result of applying overlay might consist of two instances, one for each MUB.	178
8.1	Basic architecture of the integrated Fusion And Discourse Engine (FADE).	180
8.2	The upper-level of FADE’s internal taxonomy for Acts	181
8.3	The functional architecture of FADE. The left part of the figure shows the conversational context that is accessed and updated by PATE rules and the right part shows the discourse context and its API for processing propositional contributions.	186
8.4	End-of-Turn detection and synchronization of incoming monomodal events.	193
8.5	FADE’s high-level processing strategy for the contextual interpretation of the propositional content of a user contribution.	194
8.6	Example configuration of a turn-sequence of four turns involving three speakers.	198
8.7	Example configuration of the physical environment describing the football field with three players placed on it.	207
9.1	Screen-shot of the VIRTUALHUMAN system during the first game phase with three virtual characters present. From left to right: The moderator, the virtual football expert Mr. Kaiser and the virtual football expert Ms. Herzog.	219
9.2	The studio during game stage two with two virtual characters. The lower part of the picture shows the football field where the line-up is visualized.	220
9.3	Basic Architecture of the VIRTUALHUMAN system.	224
9.4	Architecture of the User-CDE and Character-CDEs of the VIRTUALHUMAN system.	225

9.5	Overview of the Act hierarchy of VIRTUALHUMAN.	226
9.6	Example rule for identifying the intended addressee in the VIRTUALHUMAN system.	230
9.7	Example of an interrupting gesture: The female character Ms. Herzog wants to take the turn and signals this to the male character—who is currently holding the turn—by raising her right arm into the gesture space.	231
9.8	The basic architecture of the SMARTWEB system.	232
9.9	SMARTWEB in action: The screen-shot on the left depicts the summary of the user’s question, the screen-shot in the middle depicts the system response and the screen-shot on the right depicts a detailed view of an answer.	234
9.10	Example interaction with the SMARTWEB system: The user points at a player displayed on the screen while uttering: “ <i>How many goals did this player score?</i> ”. 235	
9.11	A SmartMedia instance representing the decomposition of a picture of the 1998 Brazil World Cup football team.	238
9.12	Two still images illustrating the function of the OnView/OffView recognizer: The image on the left shows the OnView case and the one the right shows the OffView case.	240
9.13	Example of the internal representation of the attentional state for SMARTWEB. 240	
9.14	The OMDIP system in action: The user selects a <i>Soundlogo</i> using the pen while uttering the request to play the selected Soundlogo.	242
9.15	The basic architecture of the OMDIP system.	244
9.16	OMDIP: A StartOfRecognition event on top of the goal-stack.	249
9.17	OMDIP: Configuration of the working memory and the goal-stack just before a pointing gesture is processed.	250

List of Tables

2.1	Examples of dialogue acts as they occur in task-oriented dialogues.	17
3.1	Different types of human-human discourse and their characteristics.	32
3.2	Different types of human-computer interaction and their characteristics. . . .	33
3.3	Attributes of the different conversational roles from the perspective of the speaker (adapted from Bell (1984, p.159)).	35
3.4	The two discourse fragments comprise exactly the same sentences but in a different order; as a result they tell two completely different stories (adapted from Russell and Norvig (1995, p. 717)).	36
3.5	Examples of Initiative-Response units (Ahrenberg et al., 1991).	40
3.6	The five hierarchical levels of spoken interactions defined by Stenström (1994, p. 30).	40
3.7	Taxonomy of bridging references as they occur in natural discourse (adapted from Clark (1977, pp. 170-172)).	42
3.8	Characterization of the three main frames of reference (Levinson, 2003). . . .	49
3.9	Givenness hierarchy for noun-phrases (Gundel et al., 1993).	52
3.10	Speech-Act types of <i>message-type</i> non-sentential utterances (fragments); β is the fragment and α the utterance to which it is related (adapted from Schlangen (2003)).	56
3.11	Continued: Speech-Act types of <i>message-type</i> non-sentential utterances (fragments); β is the fragment and α the utterance to which it is related (adapted from Schlangen (2003)).	57
3.12	Continued: Speech-Act types of <i>message-type</i> non-sentential utterances (fragments); β is the fragment and α the utterance to which it is related (adapted from Schlangen (2003)).	58
3.13	Examples of communal groups and their shared expertise (adapted from Clark (1996)).	59
3.14	The seven <i>Grounding Acts</i> of Traum's <i>Grounding Acts Model</i> (Traum, 1994). . . .	60
3.15	Some examples of common lexical items that serve as interactional signals or discourse markers in various contexts and their functions (adapted from Stenström (1994, p.66)).	61
4.1	Input synchronization patterns used for a multimodal mobile shopping assistant (adapted from Wasinger et al. (2005)).	104
8.1	Overview of the different strategies for the resolution of different types of referring expressions.	199

8.2	Examples of temporal deictic expressions and the corresponding offset in days or hours.	204
8.3	Examples of time deictic expressions and the corresponding offset in hours and minutes.	204
8.4	Some dates for dynamic and fix holidays in Germany.	205
9.1	Examples of communicative functions and their behavioral realizations in VIRTUALHUMAN.	231
9.2	Overview of the individual discourse phenomena as they are covered by FADE in the VIRTUALHUMAN, SMARTWEB and OMDIP systems.	251
9.3	Overview of FADE's rule base for the SmartWeb, VirtualHuman and OMDIP systems.	251
9.4	Overview of FADE's processing times for a selection of sample input taken from the SMARTWEB system.	252

Chapter 1

Introduction

For a long time now, human-computer interaction has been dominated by artificial human-computer interfaces like the computer keyboard or the mouse. While these devices require the user to adapt herself or himself to the interface and the application in order to be able to use it, there have, in recent years, been developments of more sophisticated human-computer interfaces. These support a more natural interaction so that it is now the system that adapts itself to the user and not vice versa. These interfaces are summarized by the term *multimodal dialogue systems*. Multimodal means that the interface supports more than one input modality so the system can be addressed, e. g., via speech input, gesture input, handwriting or facial expressions. Moreover, these systems also support a variety of output modalities such as speech, gestures or facial expressions. This variety of input and output channels offers a greater flexibility for the user when interacting with the system.

With the advent of multimodal dialogue systems, a number of new research topics emerged. One of them focuses on the contextual integration and interpretation of the monomodal actions of the users into a coherent representation of her or his intentions. The main focus of this thesis is the development of a comprehensive context model for multimodal multiparty dialogue systems that supports the integration and interpretation of multimodal contributions in the light of their context. Another important aspect of this thesis is the development of a generic component that realizes an integrated fusion and discourse processing component based on this context model.

A further goal of this thesis is to contribute to bridging the gap between commercial and research dialogue systems. This means that the work presented here is guided by the idea of developing a generic toolkit that is easy to adapt to new systems or applications. Efficiency, robustness, flexibility and promptness of reaction are hereby considered to be the key requirements for the developing process. Thus, this thesis touches three fields in the area of multimodal dialogue systems: (i) multimodal fusion, (ii) context modeling and multimodal discourse processing and (iii) bridging the gap between commercial and research dialogue systems.

1.1 Multimodal Fusion

In contrast to only speech-based dialogue systems, multimodal dialogue systems encompass a number of input modalities that can be employed by the user in an isolated or combined way in order to interact with the system. A classic example of a multimodal utterance is

a spoken command like “*Play this song*” that is accompanied by a pointing gesture thereby selecting a specific song displayed on the screen (e. g., by using a pen). In order to be able to trigger an appropriate reaction to such an utterance, a multimodal dialogue system needs to integrate the two monomodal actions of the user into a coherent multimodal interpretation of their intention. This task is usually carried out by a component called *multimodal fusion*.

In general, the task of a modality fusion component is to combine and integrate—if possible—all incoming unimodal events ($e_0 - e_n$) into a single representation of the intention most likely expressed by the user. A fusion component has to ensure that every unimodal event that could potentially contribute to the integrated meaning of a multimodal utterance ($m(e_0, \dots, e_n)$) is considered. Thus, a fusion component needs to synchronize the recognition and analysis components so that all unimodal components of an utterance will be taken into account. The opposite process of multimodal fusion is modality fission (or turn-planning) where an abstract representation of the content that is supposed to be communicated to the user has to be reasonably distributed over the available modalities.

1.2 Context-based Interpretation of Multimodal Utterances

Previous research on context modeling and discourse processing has mostly focused on two-party discourse (see, for example, Luperfoy (1991); Bunt (1994); Poesio and Traum (1995); Bunt (2000); Salmon-Alt (2000); Allen et al. (2001)). It has mainly been concerned with aspects like maintaining a coherent history of an evolving spoken discourse in order to support the resolution of referring and elliptical expressions. Nonverbal behavior as well as multiparty effects play only a minor role and are addressed in distinct approaches to multimodal fusion that do not incorporate contextual information. However, it has become clear that the existing approaches are not sufficient when we start considering the complete variety of individual phenomena occurring in multiparty human-human conversations (e. g., turn-taking management, backchannel feedback, addressee identification). Existing discourse models lack the ability to represent and store a wide number of specific events (e. g., eye movement, body posture or facial expressions) but more importantly they are not equipped to interpret those events with respect to the dialogue situation in which they are uttered.

Contextual information influences the understanding and generation of communicational behavior (see, for example, Duncan (1972), Sacks et al. (1974), Bunt (2000)) and it is widely acknowledged that any multimodal dialogue system dealing with more or less natural input must incorporate contextual information. The primary goal of this thesis is to develop a generic and comprehensive context model that supports the integration of perceived monomodal events into a multimodal representation, the resolution of referring expressions and the generation of reactive actions. Besides the classic linguistic context provided by a discourse history, we have identified sets of physical and conversational context factors that a multimodal dialogue system aiming at real conversational interaction needs to incorporate. Using this extended notion of context, we aim at processing both verbal and nonverbal behavior in dyadic and multiparty conversations and for that purpose an integrated approach to multimodal fusion and discourse processing called FADE will be developed.

1.3 Bridging the Gap between Commercial and Research Dialogue Systems

Despite their commercial success, current commercial dialogue systems show only limited capabilities with regard to natural dialogue. Many of these systems do not support the resolution of referring or elliptical expressions and require reduced and simplified user commands. Moreover, true mixed-initiative dialogue management is still only available in research prototypes and users have to deal with system-driven dialogue management. Since the overall goal of employing a speech dialogue system is to ease the interaction with computers, more powerful dialogue systems are needed so that users do not need to adapt their natural conversational behavior to that of the dialogue system.

While VoiceXML-based dialogue systems¹ typically consist of a set of well-defined VXML documents that are processed by a VXML-interpreter, state-of-the-art systems consist of a number of components of which each one typically comes with its own knowledge base. This means that the price for enhanced functionality and coverage of such systems is their general complexity and reduced maintainability.

1.4 System Context

The system context for this thesis comprises three distinct multimodal dialogue systems: (i) VIRTUALHUMAN, (ii) SMARTWEB and (iii) OMDIP. What all three systems have in common is that they employ an ontology-based data representation (i. e., every concept the system can understand or talk about is represented by means of a system-wide ontology). Moreover, all three systems support a variety of discourse phenomena whose interpretation requires contextual knowledge that goes beyond a simple dialogue history.

The integration of FADE as the central component for dealing with context information in these systems serves as a proof of concept for this approach.

VirtualHuman

This work has been conducted as part of the research project VIRTUALHUMAN² (Reithinger et al., 2006; Göbel et al., 2006). VIRTUALHUMAN is a long-term research effort aiming at the development of virtual characters that act as comprehensive, life-like dialogue partners. The emphasis of this project lies on achieving a highly realistic graphical representation of both the virtual environment and the characters and also on a natural interaction metaphor by means of an affective multimodal human-computer interface. The virtual characters have to respond to contributions made by the human users and that is why it is necessary for them to be able to act autonomously and with respect to the situational context. The actual verbal and nonverbal behavior of the virtual characters is controlled by means of *Conversational Dialogue Engines* (CDEs). Each CDE autonomously plans and executes the individual actions of a single virtual character and comprises, besides other components, an instance of FADE, an action planner and a multimodal generator.

The first application scenario of the VIRTUALHUMAN system is a school lesson where a virtual teacher teaches both a virtual pupil and a human pupil—the user—in astrophysics

¹See <http://www.w3.org/TR/voicexml20/>.

²See <http://www.virtualhuman.de>.

(see figure 1.1 for a screen-shot). This scenario served as a proof of concept for the successful integration of the various sub-components of VIRTUALHUMAN but had only limited dialogue capabilities.

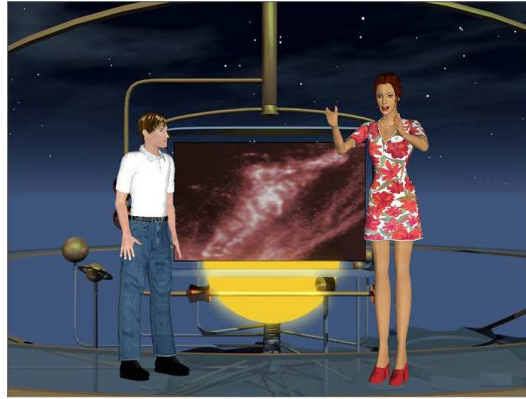


Figure 1.1: Screen-shot of the first version of the VIRTUALHUMAN system in action. The teacher (on the right) teaches both the pupil (on the left) and the human user, who is in front of the system.

The second and final demonstration scenario is a quiz-game show that is situated in the context of the 2006 Football World Cup in Germany. The quiz is hosted by a virtual moderator. Besides two human users, there are two virtual experts to help the human contestants (see figure 1.2). The game has two stages: In the first stage, the moderator shows short videos of suspenseful situations from past football games³. These videos stop just before the situation is about to be resolved (e. g., the striker attempts to shoot) and the human contestants have to guess what will happen next. For each right answer, the contestants can collect points so that at the end of the first stage a winner can be pronounced. The winner then proceeds to the second stage where the user has to work with the moderator and one of the experts to assemble a team lineup of the German national football team (see the right part of figure 1.2).

A key feature of VIRTUALHUMAN is that the users can interact with up to three virtual characters in a natural way. However, this poses high demands on the dialogic capabilities of the system. In order to be able to engage in a natural interaction with the users and the other virtual characters, a character needs to follow a number of conversational rules that define who is supposed to speak, or when to speak. Also, a virtual character that perceives an utterance needs to be able to identify whether it has been addressed by this utterance or not. Moreover, in order to be able to interpret and generate referring expressions, the virtual characters need to maintain a detailed representation of the ongoing interaction encompassing all concepts mentioned in the previous discourse. FADE is used in the VIRTUALHUMAN system to realize all context-dependent aspects of the system.

³Note that we use the term *football* to refer to what in American English is called soccer.

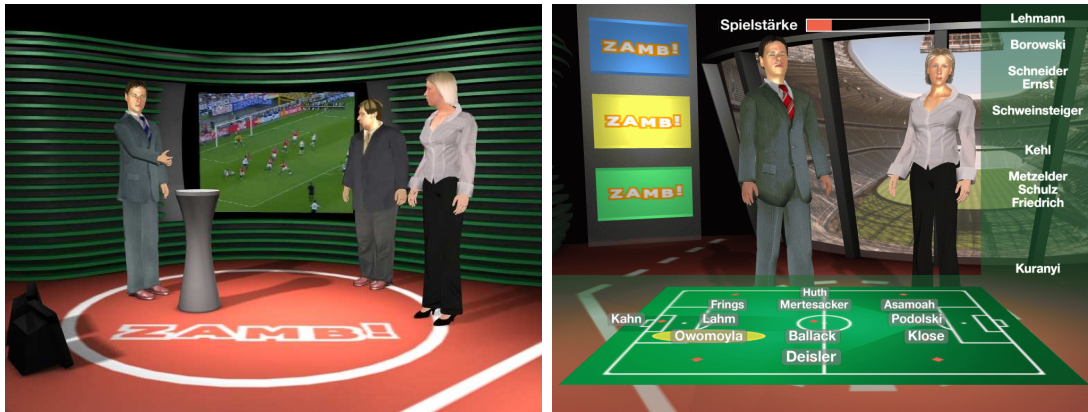


Figure 1.2: The left part of the figure shows the second version of the VIRTUALHUMAN system during the first game phase. The right part shows game phase two.

SmartWeb—Multimodal Access to the Semantic Web

The goal of the SMARTWEB project⁴ (see Wahlster (2004b); Reithinger et al. (2005a)) is the development of an intelligent mobile information service for various domains. The SMARTWEB system integrates question answering and ontology-based information extraction within a multimodal dialogue system.

The user is able to ask open-domain questions to the server-based dialogue system via a smart-phone. Primary input modalities are speech and pen input that operate in a continuous recognition mode. There is also a camera based on-/off-view detection and a prosody-based on-/off-talk detection that prevent the system from processing user contributions that were not intended to address the system. Figure 1.3 depicts an example configuration of the graphical user interface of the SMARTWEB system.

An important aspect of SMARTWEB is its context-aware processing strategy. All recognized user actions are processed with respect to their situational and discourse context. The users are thus not required to pose separate and unconnected questions. In fact, they can refer directly to the situation, e. g., “*How do I get to Berlin from here?*” Here is in this case resolved via GPS information. The user can also refer to the discourse context, e. g., in the context of a previously posed question (“*Who won the World Cup in 1990?*”) the user can pose a follow-up question (“*And in 2002?*”). Furthermore, the system can also deal with referring expressions like in the following sample dialogue:

- (1) **User:** “*Who won the World Cup in 2002?*”
System: “*Brazil.*”
User: “*How often did this team win the World Cup?*”

With the referring expression *this team*, the user refers to the discourse entity introduced by the system in the previous turn. Finally, SMARTWEB also features a variety of combined multimodal input where the spoken utterance is accompanied by pen-gestures.

⁴See <http://smartweb-project.de>.



Figure 1.3: Sample interaction with the SMARTWEB system: The left part depicts the result of the user request “Who won the World Cup in 2002?”; the right part depicts the result of the follow-up question “How often did this team win the World Cup?”

OMDIP

The OMDIP project is an industry project that serves as a proof of concept for the successful transfer of research results into a commercial application.

The demonstration scenario of OMDIP is a multimodal web-based platform for selling *Soundlogos*. Soundlogos replace the normal ring tone that callers hear until the callee picks up the phone. With soundlogos, it is possible to replace the ring tone with complete recordings of popular songs. The idea of the OMDIP application is to provide the users with an interface that enables them to change their current Soundlogo themselves. In order to change the Soundlogo, several sub-actions have to be supported by the dialogue system: Listening to Soundlogos, asking for price information and activating a Soundlogo.

The use-cases of OMDIP focus on reference resolution, multimodal integration and flexible dialogue management. Consider, for example, the following excerpt from an interaction with the OMDIP system⁵:

- (2) **System:** [displays overview of Soundlogos as depicted in figure 1.4]
User: “I want to listen to this.” [pointing gesture at *Perfekte Welle*]
System: [plays the Soundlogo *Perfekte Welle*]
User: “How much is it?”
System: “The Soundlogo ‘Perfekte Welle’ is 1.99 Euro.”
User: “Please activate it.”

⁵English translation of the German original dialogue. Note that throughout this thesis all sample dialogues of the three dialogue systems will be given in English even though the system language in all three cases is German.



Figure 1.4: The OMDIP system in action: The user selects a *Soundlogo* using the pen while uttering the request to play the selected Soundlogo.

System: *“The Soundlogo ‘Perfekte Welle’ has been activated. What do you want to do next?”*

1.5 Main Objectives

This thesis contributes to the fields of multimodal discourse processing, multimodal interaction and context modeling. The main objectives of this work encompass two primary issues: First of all, we aim at developing a comprehensive context model for multimodal multiparty dialogues that covers a variety of context-dependent discourse phenomena. Second of all, we are concerned with the requirements for a generic and reusable component that can be used for a variety of applications and dialogue systems.

Based on these two top-level research objectives, a number of research questions evolved that are addressed in this thesis:

- **What are the relevant verbal and nonverbal actions that contribute to the context of multimodal multiparty dialogue systems?** The two key aspects of human communication are verbal and nonverbal actions. Whenever speakers contribute to a conversation, they not only express themselves verbally but also by means of nonverbal actions. By considering the key characteristics of verbal and nonverbal actions, we lay the basis for an integrated processing of the total communicative situation.
- **What are the relevant factors that define the context of a multiparty human-computer dialogue systems?** Previous work on context modeling encompasses a

number of context models for multimodal dialogue systems. Even though they have been employed successfully, most of these models fail to cover the complete range of multimodal actions. Moreover, most of the approaches do not incorporate the participants or other aspects of the conversational context. We will therefore investigate to what extent the physical environment, physical actions and ontological world knowledge can be incorporated into the context model.

- **How can reactive behavior be realized in multimodal dialogue systems?** Participating in a multiparty discourse requires for a wide range of reactive behavior. Among other tasks, turn-taking and backchannel feedback require for a fast interpretation of the individual signals displayed by the other participants and a reactive display of appropriate feedback. In this thesis we will develop a model of the conversational context that realizes the reactive behavior of an artificial participant of multiparty interactions in combination with a production rule system.
- **What type of architecture is required to support the resolution of multimodal and cross-modal referring expressions in a generic way?** The usage of multimodal and cross-modal referring expressions is a key aspect of connected and situated discourse. But since the resolution of some referring expressions requires access to contextual information that goes beyond the scope of a simple discourse history, a more elaborated approach to discourse processing is needed. In this thesis we will develop an approach to discourse processing that supports the resolution of a variety of referring expressions in a generic way.
- **What is a reasonable trade-off between a comprehensive representation of meaning and efficient data handling for discourse processing?** Representation of meaning in multimodal dialogue system has a long research tradition. However, the efficient handling of complex ontological data is still an open issue. Since one of the main objectives of this work is to bridge the gap between commercial and research systems, it is of particular importance to develop an efficient but generic representation format and internal model for representing and manipulating meaning that supports various input formats.
- **Is there a way to integrate the usually separated multimodal fusion and discourse processing components?** Usually, multimodal fusion and discourse processing are realized by means of two distinct components that sequentially process user contributions. A key issue of this thesis is to develop an integrated approach to multimodal fusion and discourse processing that supports the processing of a variety of context-dependent multimodal discourse phenomena.

1.6 Thesis Outline

This thesis is divided into three parts: The first part (*Theoretical Background*) lays the basis for this work by reviewing the key context-dependent characteristics of human-human communication. The second part (*Related Work*) provides a detailed discussion of related work in the area of multimodal dialogue systems and knowledge representation. The last part (*An Integrated Approach to Multimodal Fusion and Discourse Processing*) introduces

the context model and the generic fusion and discourse engine that have been developed within this work.

A brief outline of the individual chapters is as follows:

- **Chapter 2: Human Communication: Verbal and Nonverbal Behavior**

Verbal and nonverbal actions are the central elements of human communication. Every contribution to a discourse consists at least of a nonverbal or verbal action or a combination of both. This chapter will deal with the basic human verbal and nonverbal actions as they are typically displayed by participants of conversations.

- **Chapter 3: Characteristics of Discourse**

The goal of this chapter is to provide an overview of the key aspects of the organization and structure of conversations between humans. We will thereby focus on those aspects of discourse whose interpretation requires access to the context of use. The primary basis for these considerations are empirical results from psychology, socio-psychology, psycholinguistics and linguistics. The considerations made in this chapter will form the basis for the work presented in the subsequent chapters.

- **Chapter 4: Computational Approaches to Multimodal Fusion and Discourse Modeling**

This chapter will focus on related work in the area of multimodal dialogue systems and will provide an overview of computational approaches to multimodal fusion, context modeling and reference resolution for practical multimodal dialogue systems.

- **Chapter 5: Representing Meaning in Multimodal Dialogue Systems**

Choosing a well suited format and internal model for representing meaning in multimodal dialogue systems is a key aspect in the design process of such applications. Since one of the overall goals of this work is to develop a generic and reusable component for multimodal fusion and discourse processing, a lot of thought has been put into this question. In this chapter the foundations of a number of recent knowledge representation approaches suitable for multimodal dialogue systems will be discussed.

- **Chapter 6: A Comprehensive Context Model for Multimodal Multiparty Discourse**

This chapter will introduce and discuss a comprehensive context model for multimodal multiparty dialogues. This model is designed to support both the task of multimodal fusion and the task of discourse processing in a multimodal dialogue system. It consists of two types of context representations: (i) the immediate conversational context and (ii) the long-term discourse context.

- **Chapter 7: PATE—A Production Rule System Based on Typed Feature Structures**

In this chapter we will discuss the central aspects of the production rule system PATE which serves as the central control instance of FADE. The PATE system provides a comprehensive framework for realizing both the generic and domain-specific aspects of a processing logic. Moreover, PATE also incorporates an efficient implementation of a knowledge representation that combines the key ideas of typed feature structures with a model of unique entities as it is used by ontologies.

- **Chapter 8: An Integrated Fusion and Discourse Engine**

While the previous chapters discuss the notion of a comprehensive context model and introduce with PATE a framework for developing components for multimodal dialogue systems, this chapter will show how these parts are incorporated into the FADE system. FADE is a generic and reusable engine for multimodal fusion and discourse processing.

- **Chapter 9: Applying FADE in Multimodal Dialogue Systems**

In this chapter we will show how FADE has been employed in three practical dialogue systems with varying requirements and capabilities. Besides showing the general operability of FADE, the purpose of this chapter is also to illustrate how FADE can be adapted to new tasks.

- **Chapter 10: Conclusion and Discussion**

This chapter aims at summarizing the work done in this thesis. It highlights again the major scientific contributions and underlines the importance of the findings made in this thesis. This chapter concludes the thesis by providing directions for future research.

Chapter 2

Verbal and Nonverbal Behavior

Verbal and nonverbal actions are the central elements of human communication. Every contribution to a discourse consists at least of a nonverbal or verbal action or a combination of both. This chapter deals with the basic human verbal and nonverbal actions as they are typically realized by participants of conversations. Section 2.1 introduces the basic characteristics and terminology regarding human communication. Sections 2.2 and 2.3 discuss the key concepts of verbal and nonverbal actions, respectively.

2.1 Introduction

Human communication is used for actions, e. g., conducting business, instructing students, negotiations or discussions. Typically, this is realized by means of an exchange of spoken utterances. However, detailed analysis of interactions between humans has revealed that even though speech is the primary mode of communication, there is a great deal of information conveyed through gestures, facial expressions or body posture. (see Goodwin (1981); Rutter (1984); Kendon (1987); McNeill (1992); Carlett et al. (2002); Knapp and Hall (2002); Kendon (2004)). Moreover, under some circumstances the meaning of verbal contributions is turned into the opposite by a simple facial expression:

- (3) *That's great!* [looks exasperated; neutral voice, no stress on the final syllable]

How the individual modes interact to convey the actual meaning of this utterance becomes particularly obvious when considering combinations of these modes. For a listener who cannot see the speaker, this utterance is at least ambiguous, depending on how much impact the prosody is given. For someone who can only read a transcript of this verbal message without the benefit of additional information about the accompanying nonverbal expressions, this message would definitively have only a single, positive meaning.

Throughout this thesis, we will use the term *utterance* to refer to any verbal and/or nonverbal behavior of a dialogue participant that is perceived by the other participants as an attempt to *convey* information of some sort (Kendon, 2004). Thus, *utterance* refers to any action or ensemble of actions displayed by a dialogue participant that has a clear start- and endpoint. Accordingly, we will use the terms *verbal* and *spoken* contributions interchangeably to refer only to those aspects of an utterance that are expressed by speech and contribute to the propositional content. The latter restriction is important, as we will subsume paraverbal spoken sounds that are generated through the speech tract under the term *nonverbal*

contributions or behavior. Thus, *nonverbal* behavior refers to all kinds of gestures, facial expressions, body posture and para-verbal behavior.¹

2.1.1 Communicative Functions vs. Communicative Behavior

As Cassell (2000) points out, even though conversations look like they are governed by rules, the set of behaviors exhibited differs from person to person and from conversation to conversation. This means that we cannot consider the surface behavior alone when we attempt to build a comprehensive model of conversations. What we need is a model of high level structural elements of interactions plus their typical function in the discourse: their *discourse function*.

The need for this additional functional layer results from the fact that particular behaviors can be employed in a number of contexts to produce different communicative functions, and moreover, the same communicative function can be realized through a number of different behaviors (e.g., Cassell (2000); Pflieger and Alexandersson (2004)). This means the behavior that is actually selected for a particular discourse function depends on several contextual factors, e.g., the currently available modalities, the personal style, alignment effects to the other participants or the current behavior of other participants (e.g., Cassell (2000); Knapp and Hall (2002)). This context dependency is crucial for our work as it serves as a primary motivation for the integrated and context-based processing of verbal and nonverbal contributions.

2.1.2 Interactional vs. Propositional Content

Following (Cassell et al., 1999; Cassell, 2000), contributions to conversations can be divided into *propositional* and *interactional* information. While propositional information contributes to the content of the conversation, interactional information contributes to the regulation and organization of the conversational process. Thus, meaningful speech and gestures that complement or elaborate on the speech content both contribute to the propositional aspect of contributions. Interactional information, however, is realized by means of a range of nonverbal behavior (like head nods indicating that one is listening, gazes) and para-verbal speech (like *hmm*, or *huh*). As we will discuss in the next chapter, it regulates the transitions between speaker and listener, helps to avoid overlapping speech and supports the identification of intended addressees of a contribution.

2.1.3 Deliberative vs. Reactive Behavior

For a complete understanding of human communication, it is important to take the different levels of consciousness into account at which verbal and nonverbal behavior is elicited. Some behaviors can be controlled and thus can be suppressed while others appear at a purely unconscious level and are hard to suppress. The content of a contribution—the actual message of the utterance—is clearly based on voluntary and goal-driven cognitive processes. What many theories of speech production have in common is that a substantial amount of pre-production planning is assumed at the initial processing stage (see Eysenck and Keane (2000)). However, the subsequent processing stages (which realize the actual content, i. e., the units of sound to be uttered) are assumed to occur at less conscious levels. Many people, for

¹We will refine and detail this definition in section 2.3.

example, have serious difficulties to break with habitual gestures or behaviors they are used to displaying at specific points in conversations.

A common explanation for this phenomenon is that some parts of these behaviors become automatic as a result of prolonged practice. In cognitive psychology there exists reasonable agreement on the following criteria for automated processes (taken from (Eysenck and Keane, 2000, p. 141)):

- 1 They are fast.
- 2 They do not reduce the capacity to perform other tasks.
- 3 They are unavailable to consciousness.
- 4 They are unavoidable.

However, even though some aspects of utterances are hard to control, they are not unavoidable. For example, as Kendon (2004) pointed out, “there is always the implication that the actor is deemed to exercise at least some degree of voluntary control over any movement regarded as ‘gesture’ and what it expresses” (Kendon, 2004, p. 8). Thus, we assume an intermediate stage—a semi-conscious level—where cognitive processes take place that still meet parts of the criteria for automated processes but that to some extent underlie voluntary control (see also Löckelt and Pflieger (2005)).

Deliberative behavior is characterized by the desire to meet specific goals and it requires, in contrast to reactive behavior, elaborated reasoning. In the area of artificial intelligence, the *Belief-Desire-Intention* (BDI) architecture of Bratman (1987) has been adopted as a central model for dealing with human practical reasoning. The model centers around three mental structures of a cognitive agent: *beliefs* comprising knowledge about the agent’s surroundings, *desires* representing the goals of the agent and *intentions* representing the deliberative state of the agent, i. e., what the agent has chosen to do. In addition to these three mental states, an agent also makes use of a plan library that consists of hierarchical plans which can be used to achieve the agent’s intentions. Throughout this thesis, we will use the terms *deliberative* behavior to refer to purely conscious, controlled processes and *reactive* behavior to refer to semi-conscious processes.

2.1.4 Meaning and Context

An important characteristic of natural discourse is that the meaning of an utterance depends on the context within which it has been expressed. A referring expression (see section 3.2.3) like “*he*” or “*she*” clearly depends on the context provided by its utterance and the previous discourse. However, as Pinkal (1977) for example highlights, there are *normal* lexemes whose meaning also depends on the situational context. Consider for instance:

- (4) *Peter is tall.*

Whether or not this sentence is true depends entirely on the situational context of this utterance, i. e., when Peter’s age is known in relation to his height. The question is to what extent the meaning is defined by the lexeme and to what extent by the context of use. Jackendoff discusses a quite extreme view of linguistic philosophy that states that “there is no fixed meaning associated with linguistic expressions; rather the best one can do is catalog the contextual uses of expressions” (Jackendoff, 2003, p. 280). But as Jackendoff (2003)

continues, this view neglects the fact that the expressions must convey at least something with which the context can interact, otherwise it would not be necessary to say anything at all as it would already be determined by the context.

Barwise and Perry (1983) put forward a notion of *situation semantics* which views the meaning of a declarative sentence as a relation between utterances and described situations. Barwise and Perry (1983) exemplify this notion by means of the sentence “*I am sitting.*” The meaning of this sentence is given by a relation “that holds between the utterance u and a situation e if there is a location l and an individual a such that in u , a is speaking at l , and in e , a is sitting at l ” (Barwise and Perry, 1983, p. 19). In the notation that Barwise and Perry (1983) introduce this is expressed as:

$$u \text{ [[IAMSITTING]] } e$$

iff

There is a location l and an individual a , such that

in u : at l speaks, a ; yes

in e : at l sits, a ; yes.

As Barwise and Perry note, the extensions of this relation will be a large class of pairs of abstract situations. However, if we fix certain aspects about the context of the utterance (e. g., Peter is the speaker and the location is a specific spacetime region l) then any situation where Peter is sitting at l will be an interpretation of that utterance. In this thesis, however, we will be not concerned with the contextual interpretation of lexemes.

2.2 Characteristics of Verbal Contributions

Even though nonverbal behavior has a great impact on the propositional content of an utterance, speech is the primary mode to communicate the propositional content. In this section we will discuss the key characteristics of verbal contributions that relate to the dialogue phenomena discussed in this thesis. We will particularly focus on context-dependent aspects of conveying meaning and will have a brief look at conversational implicature, speech act theory and dialogue acts.

2.2.1 Conveying Meaning—The Propositional Content of an Utterance

The question of how meaning is conveyed in an utterance can only be fully answered when we differentiate between two related subfields of linguistics: (i) *pragmatics* and (ii) *semantics*. While the former is concerned with the context-dependent interpretation of an utterance, the latter is concerned with its linguistic, context-independent meaning of the sequence of words in terms of truth values (as mentioned in the previous section). We will be concerned with the proposition of concepts and with the interpretation they receive when they are uttered in the context of a dialogue. Thus, in what follows we will focus on the pragmatic effects on the propositional content of an utterance. In the remainder of this section we will briefly discuss the most influential treatments of context-dependent interpretation of dialogues.

2.2.2 Conversational Implicature

Grice (1975) put forward the theory of *conversational implicature* which describes a basic set of licensed inferences that the speaker and the listeners can expect each other to follow. Grice (1975) transferred these inferences into a set of **maxims** which play a guiding role in the interpretation of utterances:

- 1 *Maxim of Quantity*—Be as informative as required.
- 2 *Maxim of Quality*—Your contribution has to be true.
- 3 *Maxim of Relevance*—Be relevant.
- 4 *Maxim of Manner*—Be perspicuous.

The key argument of Grice (1975) is that communication will be successful and efficient as long as both the speaker and the listener(s) follow these maxims. For instance, consider the following utterance:

- (5) **A**: “*Peter owns five cars.*”

Here, **A** states that *Peter* owns exactly five cars, even though this sentence would be true if *Peter* owns six or more cars. Because of the maxim of quality any listener infers that **A** provided the exact amount of cars.

2.2.3 On Speech Act Theory and Propositions

Austin (1962) introduced another interesting insight into conversation, as he noticed that speakers perform *actions* with each utterance. A good example for these actions are *performative* sentences like:

- (6) *I pronounce you husband and wife.*

When uttered by the appropriate authority this sentence will have the effect of changing the state of the world (i. e., it causes the two people addressed to be married). This change or *action* is basically provoked by the verb of that sentence and Austin (1962) called these kinds of actions *speech acts*. According to the speech act theory, each utterance in a communicative situation constitutes three kinds of acts:

Locutionary act—The utterance itself, i. e., its meaning and reference.

Illocutionary act—The function of the utterance as it will be understood by the hearers.

Perlocutionary act—The effect that this utterance has on the addressee(s).

As the illocutionary act is the most obvious act of these three, the term speech act is now generally used to describe this act rather than all three of them. A special condition called *indirect speech acts* occurs when the literal illocutionary act does not match the primary or intended illocutionary act. An utterance like “*There is the door*” is actually a declarative statement that is supposed to realize an assertion. Under certain conditions, however, this utterance is understood as a request to leave the room. Searle (1969), a student of Austin’s, builds upon Austin’s work by stating that a speaker uttering a sentence is typically performing three distinct kinds of acts (Searle, 1969, p. 23):

- 1 Uttering words (morphemes, sentences) = performing *utterance acts*.
- 2 Referring and predicating = performing *propositional acts*.
- 3 Stating, questioning, commanding, promising, etc. = performing *illocutionary acts*.

Searle (1969) also distinguishes two types in the syntactical structure of sentences which he calls the *illocutionary force indicator* and *propositional indicator*. The illocutionary force indicator shows how the proposition has to be taken. As Searle (1969) summarizes, illocutionary force indicators in English can be word order, stress, intonation contour, punctuation, the mood of the verb and performative verbs. However, as Searle (1969) notes, in actual speech situations it is often not necessary to invoke the illocutionary force indicator directly since the context will make clear what the illocutionary force of the utterance is. In (Searle, 1975), he extends these ideas and proposes that all speech acts can be classified into one of five categories:

- 1 *Assertives*—utterances that may be judged true or false.
- 2 *Directives*—utterances that attempt to make the addressees' actions fit the propositional content.
- 3 *Commissives*—utterances committing the speaker to a course of actions.
- 4 *Expressives*—utterances that express the speaker's attitude or psychological state.
- 5 *Declarations*—utterances that change (or attempt to change) the state of the world.

2.2.4 Dialogue Acts

Based on the work of Austin and Searle, a lively research community has arisen that focuses on modeling several other kinds of conversational functions of utterances. As a consequence, a variety of approaches and terms have been put forward to enrich speech acts with additional functions. Among the most prominent approaches are: *communicative acts* (e.g., Allwood (1976)), *conversational acts* (Traum and Hinkelman, 1992), *dialogue acts* (Bunt, 1994) and *conversational moves* (Carletta et al., 1997). Here, we will only briefly discuss *dialogue acts* since it is one of the most influential approaches.

The idea of dialogue acts originates from Bunt's *Dynamic Interpretation Theory* (DIT) (Bunt, 1994, 2000). Key to this theory is the idea of a *context-changing* aspect of utterances which is expressed through *dialogue acts*. As it is pointed out in (Bunt, 2000), the meaning of an utterance can be viewed in terms of context changes by distinguishing between the information the speaker is introducing into the context (*semantic content*) and the way information has to be inserted into the context in order to play the intended role (*communicative function*). The combination of semantic content and communicative function of an utterance results in the *dialogue act* of an utterance. Dialogue acts are often organized in a *hierarchical* way so that there are generic and specialized dialogue acts. Table 2.1 displays some examples of dialogue acts as they tend to occur in task-oriented dialogues.

A lot of effort has been put into the development of standardized dialogue act taxonomies, e.g., the Verbmobil dialogue act taxonomy, (see Alexandersson et al. (1998)) or the DAMSL (Dialog Act Markup in Several Layers) approach (e.g., Carletta et al. (1997)). However, as pointed out in Traum (2000), this is a very complex process and often subject to system or

Dialogue Act	Description
STATEMENT	a claim made by the speaker
REQUEST	a question by the speaker
RESPONSE	a response to a previous request
OPENING	greeting

Table 2.1: Examples of dialogue acts as they occur in task-oriented dialogues.

task-specific compromises. The dialogue systems that set the system context for this thesis all make use of dialogue acts.

2.2.5 The Context Changing Function of Verbal Contributions

As we will discuss in the next chapter there are a variety of factors inherent to language like referring expressions that can only be resolved by considering their context. Thus, *context* serves as some kind of additional input for the interpretation of these phenomena but the interpreted utterance also modifies this context. In that sense, a modified representation of context is also the output of the interpretation process. As discussed in Bunt (2000), this notion of context change is closely related to the notion that language is interaction as it is stated by the speech act theory (see above).

When we construe context to be the totality of conditions that influence the understanding and generation of linguistic behavior, and note that a communicative action obviously changes these conditions, we must conclude that a communicative action operates on a given context to produce a new one. (Bunt, 2000, pp. 81–82)

With his *Dynamic Interpretation Theory* (DIT), Bunt provides a formal framework for the context change aspect of linguistic actions. In DIT, utterance meanings are viewed in terms of context changes. An utterance is taken to consist of a *semantic content* representing the information the speaker inserts into the context as well as a *communicative function* which is defined as the way in which dialogue participants use information to change the context². Given a semantic content p of an utterance, its communicative function is represented as a context update function $F(p)$ that computes—based on a given context Γ —an updated context Γ' .

2.3 Characteristics of Nonverbal Contributions

Usually when people refer to the term *nonverbal behavior*, they mean the visible hand and arm movements that speakers perform in contrast to speech. Nonverbal behavior, however, is more than gestures and the purpose of this section is to outline the notion of nonverbal behavior that we adopted for this thesis. To this end, we will discuss a detailed classification of different types of nonverbal behavior.

²These notions are closely related to the concepts of propositional content and illocutionary force of the speech act theory.

2.3.1 Classifying Nonverbal Behavior

When we try to classify nonverbal behavior, the first thing to do is to consider it in contrast to verbal behavior. This is a common view, as (Knapp and Hall, 2002, p. 5) have put it: “[t]o most people, the phrase *nonverbal communication* refers to *communication effected by means other than words*.” However, as Knapp and Hall continue, this definition is somewhat problematic as it is virtually impossible to separate verbal and nonverbal behavior into two distinct categories. They endorse this view of inseparable categories by means of American Sign Language which is the language by which deaf people communicate. They argue that this gesticulation is mostly linguistic and thus verbal but that hand gestures in general are viewed as being nonverbal. Bußmann (2002) defines nonverbal communication as follows (English translation of the German original): “The entirety of nonverbal phenomena occurring in interpersonal communication processes [...]. With respect to the signals of nonverbal communication we discern between (a) vocal means like volume, pitch, speech rhythm, laughter, coughing, etc. [...]; (b) non-vocal (motoric) phenomena like facial expression, gesture, body-posture, eye contact, outer appearance and clothing.”

Thus, Bußmann (2002) also bases the definition of nonverbal behavior on the contrast to verbal behavior and draws a line between vocal and non-vocal, motoric nonverbal phenomena. This means that there is still the problem with sign language. For our purposes, however, this is not too problematic as we do not consider sign languages at all. Thus, we will use the notion of Knapp and Hall (2002) where nonverbal behavior is defined as communicative behavior effected by means other than words. In the following subsections we will give a compact overview of common categories of nonverbal behavior to provide some background for the following chapters. Note that some of these categories will only be viewed very briefly depending on their impact for this thesis.

2.3.2 Gestures

The word gesture comes from the Latin word *gestura*, which in turn stems from the Latin word *gerere*, which can be translated to *bear*, *wield*, *perform*. The noun gesture itself is defined in the online edition of the Oxford English Dictionary³ as follows (Oxford English Dictionary, 2006):

- 1 A movement of part of the body to express an idea or meaning.
- 2 An action performed to convey one’s feelings or intentions.
- 3 An action performed for show in the knowledge that it will have no effect.

While meaning **3** goes in a somewhat different direction, meaning **1** and **2** highlight two important aspects of gestures: (i) expressiveness—a gesture can be used to express an idea or meaning (which refers to the semantic aspect of gestures)—and (ii) a gesture can be used to convey feelings or intentions (which refers to the pragmatic aspect of gesture). Thus, gesture can be considered as being expressive of thought or feeling (Kendon, 1987). But not every action that falls under this category is considered as a gesture. Following Kendon (1987) we need to exclude:

- 1 habitual or involuntary actions (e. g., self-grooming),

³See <http://www.askoxford.com/>.

- 2 spatial and orientational movements,
- 3 practical actions (e.g., when people are engaged in a conversation, they might perform other actions at the same time like drinking or smoking).

The common feature of all these excluded actions is that they are considered to not contribute to the interaction or propositional development of the interaction. Thus, to be considered a gesture, an action must either have clear interactional effects or contribute to the propositional content of an utterance. On an abstract level we can distinguish two types of gestures: (i) speech related—gestures that are directly related to what is being said and that cannot be interpreted on their own and (ii) speech independent—gestures that are not related to any accompanying speech and that have a self-contained meaning with a direct verbal translation.

McNeill (1992) has introduced *Kendon's Continuum*, which classifies the different types of hand gestures along a continuum of lexicalization, i. e., the extent to which they are word-like (see figure 2.1). When moving from the left to the right, the obligatory presence of speech decreases while the presence of language-like properties increases.

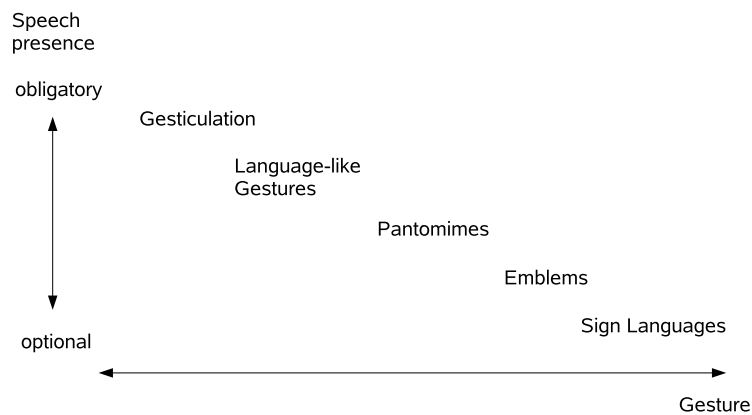


Figure 2.1: Kendon's continuum (adapted from McNeill (1992)).

Based on the relevant research literature (see, for example, McNeill (1992); Knapp and Hall (2002); Kipp (2004); Krauss et al. (1996)), we will provide a classification of naturally occurring gestures into six categories in the following subsections. It should be noted that these six categories and in particular their naming, have often been criticized by various researchers. However, for the work presented here, the actual classification is not too important as we will focus only on a smaller subset of these gesture classes and the subset will contain only those classes that are more or less accepted. The following definitions are closely related to the classification of (Kipp, 2004, pp. 39-43).

Adaptors

Adaptors are self and object touches (e. g., clothing, pencils or eyeglasses) that are considered to be non-communicative gestures, like wiping one's eye or scratching one's temple. There is an ongoing debate as to whether adaptors should be considered gestures (see McNeill (1992)). But as Kipp (2004) argues, these gestures reveal at least some information about the speaker's

or listener's mental state, for example, if he or she is in nervous or tense. Adaptors are located at the very low end of Kendon's continuum and Krauss et al. (1996) note that they tend to be considered simple hand movements and not gestures.

Emblems

The term emblem goes back to Efron (1941). Emblems are speech independent gestures that express culture-specific standardized meanings. Typically, emblems have a direct *translation* into words. The *thumbs-up* gesture, for example, stands in Western societies for a general approval of what has been said and means "yes" or "great." However, even in relatively similar societies this can mean something completely different. Figure 2.2 depicts an example of an emblematic gesture. In this example the speaker, Rudi Völler, says "[...] and I will not put up with this any longer" while performing a reproachful gesture.



Figure 2.2: Example of an emblematic gesture (from Kempe (2005)).

Deictics

The purpose of deictic gestures is to select or indicate some object, person, location or direction with respect to the physical environment (see figure 2.3 for an example). These gestures are also called pointing gestures and belong to the best-studied form of gestures. Deictic gestures are commonly considered to be made with the hands, but as Kendon (2004) stresses, they can also be expressed by using the head, the eyes or even the lips. Moreover, as Kendon (2004) stresses, there are several sub-classes of pointing gestures (e. g., *open-hand pointing*, *index-finger extended*) and he cites some studies suggesting that these differences in form may also make a difference in meaning.

Iconics

Iconic gestures illustrate or emphasize what is being said by depicting some aspect of the accompanying spoken expression. An example of an iconic gesture is if someone says:



Figure 2.3: Example of a deictic pointing gesture (from Kempe (2005)).

- (7) “A spiral staircase is a staircase going up like this.” [words accompanied by a spiraling gesture upwards with the index finger].

As Kipp (2004) notes, there is no standard form of iconics since they are often made up on the spot and this makes them very difficult to recognize and analyze for gesture recognizers. Figure 2.4 depicts an example of an iconic gesture that is accompanied by “*but you ought to come off your **high horse**.*” Here, the speaker indicates the height of the fictive horse with his hands in order to emphasize his critical statement about the other person.



Figure 2.4: Example of an iconic gesture. The speaker on the right indicates the height of an object (from Kempe (2005)).

Metaphorics

Metaphoric gestures resemble iconic gestures since they both illustrate what is being said verbally, however, they do this in a less direct way. In contrast to iconics, metaphorics employ a third element that acts as metaphor for the accompanying spoken referent. McNeill (1992) characterizes metaphorics as follows: “These are like iconic gestures in that they are pictorial, but the pictorial content presents an abstract idea rather than a concrete object or event” (McNeill, 1992, p. 14). Figure 2.5 gives an example of a metaphoric gesture. As Kipp (2004) acknowledges, it is hard to draw a sharp line between iconics and metaphorics and thus there will always be discussions about whether a gesture should be categorized as an iconic or metaphoric gesture.



Figure 2.5: Example of a metaphoric gesture (from Kempe (2005)).

Beats

Beats are rhythmic, repetitive movements with one or two hands, emphasizing the flow of the accompanying speech. Beats are often used to highlight particularly important parts of a spoken utterance and are often used as stylistic device for presentations or speeches. McNeill (1992) compares them to highlighters in written texts; their semiotic value is that they indicate the words accompanied are significant. McNeill (1992) also notes that beats tend to have the same form regardless of the content and that they are often performed as short and quick movements at the periphery of the gesture space. Displaying beat gestures also has the effect of suppressing the attempts of listeners to take the turn (see section 3.3.2).

Artificial Gestures

This gesture category cannot be found in classic gesture research literature but rather in literature about multimodal dialogue systems. Usually, such gestures are performed by means of devices (e. g., pen-based graphic tablets, touch screens, joysticks, haptic devices) or special, predefined movements of the hands employed in a similar fashion as emblems that are tracked by means of data-gloves or vision based recognition techniques.

Especially pen-based multimodal interaction has received a lot of interest in the last few years. The groundwork for the most obvious application of pen-based interaction is laid in the pioneering work of Bolt's *put-that-there* system (Bolt, 1980) and the XTRA system (Wahlster, 1991). Using the pen to select objects that are displayed on the screen is very similar to how natural deictic gestures are employed. However, as pointed out by Oviatt, pen-based interaction introduces more modes than this rather simple selection of objects (Oviatt, 1999b). Studies on integrated speech and pen input have revealed that people find it natural to employ pen input to draw graphics, symbols or signs and to write digits or lexical content (see Oviatt et al. (1997), Vuurpijl et al. (2004)). Following Oviatt et al., the *point-and-speak* pattern only comprises about 14% of spontaneous utterances (Oviatt et al., 1997).

Another instance of artificial gestures is employed for navigating in virtual reality (VR) environments. These systems often employ artificial hand gestures for navigating within the virtual reality. However, these types of gestures could be treated like emblems as they are typically associated with a standardized meaning not unlike emblems.

2.3.3 Eye Behavior

Looking into someone's eyes is called *gazing*, and this is one of the central activities for expressing attention and involvement during conversations. Where we look, when we look, and how long we look reveals a lot about our current understanding of an ongoing interaction. When two people look into each other's eyes, this is called *mutual gaze*.

As we will see later, eye gaze provides or at least supports a smooth exchange of speaking turns. Kendon (1990) provided one of the first comprehensive accounts of gaze in dyadic conversations and identified some of its fundamental functions. But more importantly, he showed that eye gaze (and also avoiding mutual gaze) has a clear communicative function. Moreover, he found that there are important individual differences in eye gaze behavior such as the proportion of the time spent looking. In the example interactions that Kendon analyzed, this varied from 28% up to 70% of the time (Kendon, 1990). Eye behavior can also be used for pointing or disambiguation purposes, e. g., a technician looking at a distinct screwdriver in a scene where more than one screwdriver lies there while saying, "*Give me that screwdriver.*" While in Western cultures it is considered rude not to look the speaker in the eye, it can be different in other cultures.

2.3.4 Facial Expressions

Facial expressions are usually considered to be the primary means through which individuals communicate their emotional state. Usually, humans can adopt facial expressions on a voluntary, deliberative basis. However, due to the close relationship between facial expressions and emotions, they are often hard or impossible to control (Knapp and Hall, 2002).

2.3.5 Vocal Behavior

Even though the heading of this section is nonverbal behavior, there is an important aspect of nonverbal behavior that is realized by speech. According to Knapp and Hall (2002), vocal behavior "deals with *how* something is said, not what is said" (Knapp and Hall, 2002, p. 10). They distinguish two high-level types of vocal behavior (Knapp and Hall, 2002, p. 10):

- 1 *Sound variations made with vocal cords during talk*, i. e., the changes in pitch, duration, loudness and silence.
- 2 *Additional sounds that result from physiological mechanisms other than the vocal cords*, i. e., pharyngeal, oral or nasal cavities.

The most interesting aspect from our perspective is the following set of nonverbal vocal cues that is part of normal speech behavior: (i) discourse markers, (ii) pauses within the speech stream and (iii) pauses during turn exchange.

Discourse Particles

A *discourse particle* or *discourse marker* (Stenström, 1994) is a lexeme or particle within a connected discourse that does not contribute to the semantic meaning of an utterance⁴. Discourse particles rather have a pragmatic function. They fill gaps, contribute to the turn-taking system or structure the speaker's relationship to the other participants. English examples are *uh, oh, uh, well, you know* and *yes*. As we will see in section 3.3.2, discourse particles have a great impact on the turn-taking systems. Duncan (1972), for example, considers discourse particles to be turn-taking signals. However, as Fischer (2000) stresses, the function of discourse particles is not entirely clear as there is no direct correspondence between a lexeme and a single function.

Paralinguistic Information in Vocal Behavior

Nonverbal vocal behavior also conveys information that allows one, for example, to estimate the age and gender of a speaker. Müller, for example, describes a two-layered computational approach called *AGENDER* that employs six different features (i. e., *pitch, jitter and shimmer, harmonics-to-noise-ratio, articulation rate, the number of speech pauses* and *the duration of speech pauses*) in order to classify speakers into eight classes with respect to their age and gender (see Müller (2006)). The first layer of the *AGENDER* system is concerned with the feature extraction and classification. This layer incorporates five different classifiers whose results are preprocessed in the second layer (e. g., by means of contextual knowledge, fusing the results of multiple classifiers with respect to one utterance to a sequence of utterances).

2.3.6 Body Posture

Body posture is considered a kind of intensifying signal that is typically studied in conjunction with other signals (Knapp and Hall, 2002, p. 9). Using posture, people tend to express attention, their social status as well as their degree of liking for the other interlocutors. A well-known example would be a forward-leaning posture which signals more liking and lower status especially when the participants do not know each other. Moreover, posture is also considered as an indicator of the intensity of the emotional state.

(Knapp and Hall, 2002) also review several studies that suggest that postural congruence serves as a source of rapport. Basically, there are two types of postural congruence that have been identified (Knapp and Hall, 2002, p. 254): (i) *matching*, a condition where the listener's behavior is exactly the same as the speaker's and (ii) *mirroring*, a condition where the listener's behavior is a mirror image of that of the speaker.

⁴Ward (2000) uses the term *conversational grunts* instead of discourse particle or discourse marker.

2.3.7 Physical Actions

Besides the nonverbal behavior that has been discussed in the previous sections, there is another category of nonverbal behavior that also contributes to the total communicative situation. We will call this class *physical actions*. Physical actions are actions performed by the participants of a conversation that directly change the state of the physical world within which the interaction takes place. However, physical actions are also part of the communicative situation since speakers can directly refer to them:

- (8) **A**: “*Could you hand me the left book?*”
 [**B** takes the book and offers it to **A**]
A: “*No, not this book; I meant the other one.*”

In this example, **A** directly refers to the book that **B** offered to **A** even though the book was never directly introduced into the discourse. In fact, it became part of the discourse because of the physical action that **B** performed.

This type of nonverbal behavior is of particular importance for tangible user interfaces (see section 4.3.2) where the user can incorporate actions involving real world objects into the interaction with the system. (Wasinger, 2006; Wasinger and Wahlster, 2006), for example, introduce the term *extra gestures* for this type of nonverbal behavior. Fleischman and Roy (2005) introduce the term *intentional actions* for those physical actions that are performed with the purpose of achieving a specific goal.

2.3.8 Functions of Nonverbal Behavior

Verbal and nonverbal actions operate together in the total communication situation in order to convey information. However, this integrated usage makes it harder to clearly distinguish between the individual functions of the two. Attempts where verbal action has been assigned to convey ideas and nonverbal actions to convey emotional information have not been very successful. Instead, “[w]ords can carry much emotion—we can talk explicitly about emotions, and we also communicate emotional messages, and we also communicate emotion between the lines in verbal nuances” (Knapp and Hall, 2002, p. 11). Argyle (1988) identifies four primary functions of nonverbal behavior in human communication:

- 1 Expressing emotion.
- 2 Conveying interpersonal attitudes.
- 3 Presenting one’s personality to others.
- 4 Accompanying speech for the purpose of managing turn taking, feedback, attention, etc.

From our perspective, the most important function of Argyle’s list is the last one since it goes in the direction of the interactional aspects of utterances that have been discussed in section 2.1. However, Argyle does not consider the fact that gestures also contribute to the propositional content of utterances. Kendon introduced a topology of four different functions of gestures that overcomes this obstacle by focusing on the interaction between the verbal and nonverbal channels (Kendon, 2004, p. 225):

- *Pragmatic function*—gestures that contribute to or constitute the acts or moves accomplished by utterances.
- *Performative function*—gestures that show what sort of a move or speech act a speaker is engaging
- *Modal function*—gestures that seem to operate on a given unit of verbal discourse and show how it is to be interpreted.
- *Parsing function*—when gestures contribute to the marking of various aspects of the structure of spoken discourse.

Kendon continues to stress the differences between these functions and the gestural form: “Any given gestural form may, according to context, function now in one way, now in another” (Kendon, 2004, p. 225). This is again an important point for our work as it highlights the importance of a contextual interpretation of perceived nonverbal behavior.

Scherer and Wallbott (1985) differentiate four basic functions of nonverbal behavior: (i) semantic, (ii) syntactic, (iii) pragmatic and (iv) dialogic. The semantic function of nonverbal behavior is either signifying referents or through amplifying, contradicting, or modifying meaning realized by verbal contributions. At the syntactical level of nonverbal behavior, Scherer and Wallbott differentiate two functions: On one hand, nonverbal behavior is used for the segmentation of utterances (e. g., eye contact or smiling to begin a conversation; change of body posture to indicate a change in the topic of the conversation). On the other hand, nonverbal behavior is used for the synchronization of displayed nonverbal behavior between participants. At the pragmatic level, nonverbal behavior reveals information about social identity, personal traits and psychological states (Scherer and Wallbott, 1985). Additionally, nonverbal behavior is often used by the listener to provide backchannel feedback (see section 3.3.4). Finally, the dialogic functions of nonverbal behavior contribute to the turn-taking system (see section 3.3.2).

Another question with respect to nonverbal behavior is to what extent it is obligatory for the understanding process of the total communicative situation. If we consider, for example, the gazing behavior of humans, it becomes clear that at least some aspects of nonverbal behavior are optional. Even though eye gaze seems to reveal important information about the turn-taking process, it is not crucial for successful communication, i. e., telephone conversations. Nevertheless, eye gaze *does* convey a lot of information as it helps to yield a turn (speaker looks at the addressee at the end of a contribution), deny it (listener looks away and does not return the look of the speaker), or display attention. For multiparty dialogue situations, Vertegaal et al. (2001) show that gaze can be used as a predictor of conversational attention. In an experimental study with 7 four-person groups discussing current affairs in a face-to-face meeting, Vertegaal et al. (2001) show that gaze can form a reliable source of input for a conversational system that needs to determine whether the user is speaking or listening to them. However, they also found considerable individual differences in gazing behavior among the subjects (e. g., standard deviations in gaze time were from 19% up to 23%).

Another aspect that clearly influences the amount of gazing is the topic or task at hand and the communication situation itself. Several studies revealed that there is also a strong decrease in gazing behavior if the task involves some visual stimuli. de Ruiter (2003) has shown in an empirical study with a map-based task that subjects gazed only 2.9 times in interactions that lasted about 123 seconds on average. However, as de Ruiter (2003) also

found “eye contact occurs almost exclusively when, to put it in informal terms, something goes wrong in the negotiation” (de Ruiter, 2003, p. 19). This means, gaze can also function as some kind of attentional indicator of joint interest if the task requires looking at the same object. But if the conversational situation requires it, participants immediately return to their original gazing behavior.

2.3.9 The Context Update Function of Nonverbal Behavior

Like verbal behavior, nonverbal behavior also changes the context of a conversation. As discussed in this section, nonverbal behavior contributes to both the interactional and propositional layer of the conversation so that a context update function for nonverbal behavior G takes as arguments the propositional content p and the interactional content i of the nonverbal action in order to update a given context Γ to Γ' . Note that for some types of nonverbal behavior p or i might be empty.

2.4 The Total Communicative Process

As discussed in the previous sections, there are speech-related gestures that are tied to or accompany speech. As a consequence, their meaning and function depend on their relation to the verbal part of the utterance. Knapp and Hall (2002) differentiate four types of speech-related gestures (Knapp and Hall, 2002, p. 241):

- 1 Gestures related to the speaker’s referent—concrete or abstract.
- 2 Gestures indicating the speaker’s relationship to the referent.
- 3 Gestures that act as visual punctuation for the speaker’s discourse.
- 4 Gestures that assist in the regulation and organization of the spoken dialogue.

Moreover, as Knapp and Hall (2002) highlight; gestures, body movements and speech are not randomly produced but they are rather “inextricably lined as parts of the same system” (Knapp and Hall, 2002, p. 250). What Verbal and nonverbal actions also have in common is that they may communicate more than one message at a time, that their meaning may be stereotyped, idiomatic or ambiguous, and that different meanings may be attributed to them in different contexts (Knapp and Hall, 2002). In the following, we will briefly discuss the basic types of interrelation between verbal and nonverbal behavior. At the end of this section we will discuss the effect of mutual disambiguation between interpretation hypotheses as they occur during natural interactions.

2.4.1 Interrelation of Verbal and Nonverbal Behavior

Ekman (1965) identifies six ways of how verbal and nonverbal behavior interrelate during interaction:

- 1 *Repeating*—The nonverbal channel simply repeats what is uttered on the verbal channel. For instance, if the speaker non-ambiguously selects an object with a referring expression but also selects the object by a pointing gesture (e.g., “Give me the left book” in the context of two books lying on a table).

- 2 *Contradicting*—Verbal and nonverbal actions do not always have the same meaning, sometimes they communicate two contradictory meanings that conflict with each other. The source of these conflicts are either problems with perception (see below) or, as Knapp and Hall (2002) discuss, mixed feelings about the communicated content, imperfect lying, or when sarcasm or irony come into play.
- 3 *Complementing*—Nonverbal actions can modify or elaborate the content of verbal utterances.
- 4 *Substituting*—Nonverbal behavior that completely substitutes the verbal message. Such nonverbal behavior stands on its own, i. e., facial expressions that provide an unambiguous picture of the emotional state.
- 5 *Accenting/Moderating*—Nonverbal behavior can be used to emphasize or moderate parts of the verbal message.
- 6 *Regulating*—Interactional contributions can be used to regulate and organize the exchange of speaking turns.

As discussed in the previous section, nonverbal behavior is often ambiguous with respect to its intended function in the total communicative process. Thus, in the concrete situation of use, there is possibly more than one way for verbal and nonverbal actions to interact with each other and this leaves the perceiver of an utterance with the problem of identifying the intended meaning of the overall meaning of the utterance.

2.4.2 Mutual Disambiguation

The interpretation of the integrated meaning of verbal and nonverbal behavior is complicated by the fact that perception is inherently uncertain. Even in surroundings with low background noise and little visual distraction, the perception and interpretation of verbal and nonverbal contributions is error-prone. This often results in a number of hypotheses reflecting possible interpretations of a single signal, meaning that the perceiver of an utterance has to determine the most promising combination of the individual hypotheses. The primary strategy for dealing with multiple hypotheses is to compare the individual interpretations of the verbal and nonverbal contributions with each other and to mutually disambiguate them until a compatible and coherent interpretation has been achieved.

In the area of multimodal dialogue systems, the process of mutual disambiguation has received considerable interest (see, for example, Wahlster (2002); Oviatt (1999a)) since the recognizers of such systems usually produce a number of interpretation hypotheses. We will take up the discussion of mutual interaction in Chapter 4 where we will discuss this issue in more detail.

2.4.3 The Context Changing Function of Utterances

Utterances in face-to-face communication are realized by means of verbal and nonverbal contributions that encompass interactional and propositional information. In order to account for the context changing function of these two types of contributions, the context update function F that extends a given context Γ must incorporate both aspects of an utterance. Thus, the context update function $F(p, i)$ takes the interactional content i and the propositional content p of an utterance to update a given context Γ to Γ' .

2.5 Conclusion and Discussion

In this chapter we have discussed key aspects of verbal and nonverbal behavior in communicative situations. We have argued that nonverbal actions should not be considered isolated phenomena, but an integral part of the total communicative process and also need to be interpreted with respect to the total communicative context. This contextual dependency is the basis for the approach presented in this thesis.

2.5.1 Our Notion of Verbal and Nonverbal Behavior

Throughout this thesis we will use the terms verbal and nonverbal behavior based on the definition given in section 2.3.1. This means that all communicative behavior that is effected by means other than words is considered to be nonverbal behavior. This classification of verbal and nonverbal behavior fits into the distinction between interactional and propositional contributions (see section 2.1.2): Meaningful verbal and nonverbal actions that complement or elaborate on the speech content contribute to the propositional aspect of contributions. Interactional information, however, is realized by means of a range of nonverbal behavior.

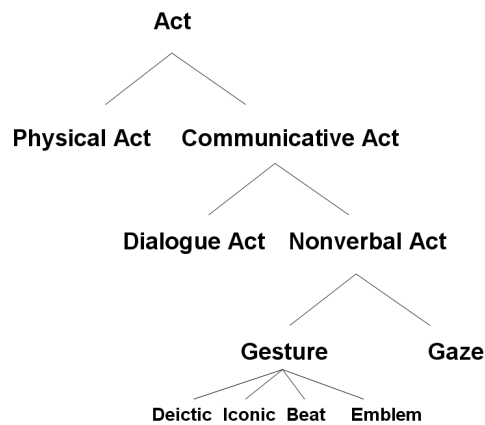


Figure 2.6: A taxonomy of verbal and nonverbal actions.

Moreover, verbal and nonverbal behavior can be integrated into a taxonomy of acts as depicted in figure 2.6. On the top-level of this taxonomy, we distinguish between *Physical Acts* and *Communicative Acts*. Communicative Acts in turn subsume *Dialogue Acts* and *Nonverbal Acts* and so on. All acts of this taxonomy interact with the total communicative context and have to be interpreted with respect to their context of use. Moreover, these acts also change context and thus have to be incorporated into a comprehensive representation of the context of an interaction.

2.5.2 Developing the Argument for an Integrated Multimodal Fusion and Discourse Processing Component

As discussed in this chapter, there are clear indications for a distinction between interactional and propositional aspects of an utterance. However, there are no indications for a disjunction

between nonverbal behavior and interactional aspects on one hand, and verbal behavior and propositional aspects on the other. In fact, the work reviewed in this chapter suggests that both nonverbal and verbal behavior contribute to interactional and propositional aspects of discourse. This is the basis for the first part of our argument for an integrated multimodal fusion and discourse processing component, since a distinction between these two components only makes sense if we are able to find a clear-cut partition in the usage of nonverbal and verbal behavior.

There is a second observation that supports our argument: Nonverbal and verbal behavior are often ambiguous with respect to their actual function in the total communicative process. This means that the interpretation of an individual action depends on its context of use and thus needs to be viewed with respect to both the immediate context settled by the encompassing utterance and the global conversational setting.

These observations also have an impact on the structure of a comprehensive context model for multimodal interactions. First of all, such a context model needs to incorporate both the verbal and nonverbal actions of the participants. But in contrast to the often applied approach of reducing the nonverbal behavior to pointing gestures, such a context model needs to incorporate the complete range of nonverbal behavior. This means that gaze behavior, posture and body orientation should be incorporated as well as the individual types of gestures and physical actions. Moreover, the distinction between interactional and propositional contributions should also be reflected by the context model. Thus, we distinguish between an immediate context model comprising the interactional contributions and a discourse context comprising the propositional contributions. Figure 2.7 gives a schematic overview of how the individual verbal and nonverbal actions contribute to these two context models.

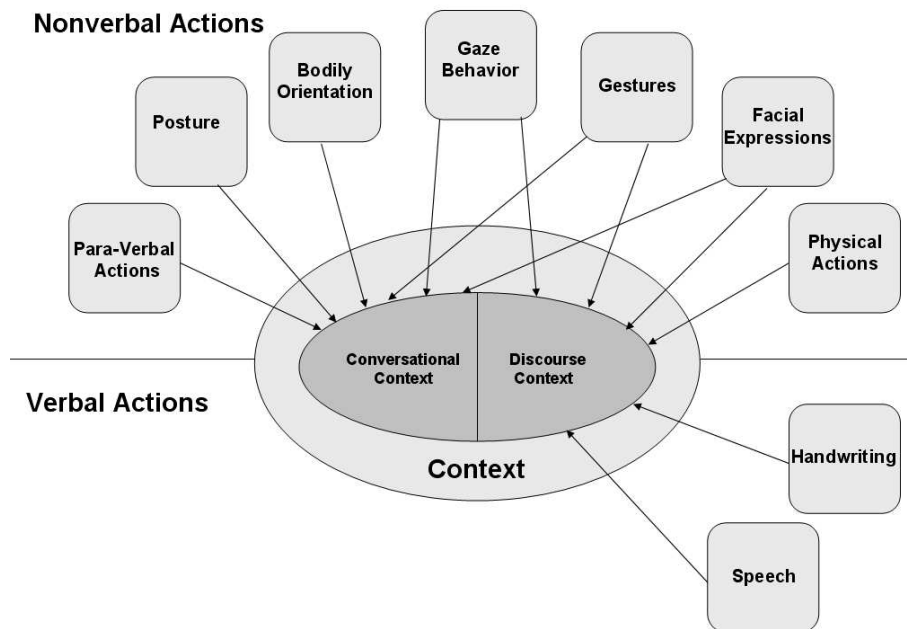


Figure 2.7: Categorization of verbal and nonverbal contributions with respect to their contribution to the context of an interaction.

Chapter 3

Characteristics of Discourse

The goal of this chapter is to provide an overview of the key aspects of the organization and structure of conversational interactions. We will thereby focus on those aspects of discourse whose interpretation requires access to the context of use. The primary basis for these considerations are empirical results from psychology, socio-psychology, psycholinguistics and linguistics and these considerations will form the basis for the work presented in the subsequent chapters. Section 3.1 introduces the basic characteristics of discourse in general and provides an overview of the different types of discourse. Section 3.2 discusses a number of context-dependent discourse phenomena that contribute to the propositional level of discourse. Finally, section 3.3 deals with context-dependent interactional aspects of discourse.

3.1 Introduction

When two or more people engage in a conversation, they need to coordinate their contributions in order to achieve a successful and smooth exchange of information. As we discussed in the previous chapter, this coordination takes place not only on a propositional level but also on an interactional level. Moreover, the more participants that take part in a conversation, the more complex the coordination of the interaction gets.

In the following subsections we will briefly discuss the different types of discourse and the different roles that participants can take in dyadic and multiparty discourse (see below). These considerations form the basis for the overview of the theoretical foundations of discourse that will follow in the remainder of this chapter. Due to space restrictions, some of the topics discussed in this chapter are not discussed in full detail. For further reading see, for example, Allen (1995); Clark (1996); Jurafsky and Martin (2000); Görz et al. (2003).

3.1.1 Types of Discourse

The term *discourse* stems from the Latin word *discursus* which means *conversation* (literally: running back and forth). The *Merriam-Webster Online Dictionary* defines the term discourse as follows¹:

- 1 *Archaic*: the capacity of orderly thought or procedure: RATIONALITY.
- 2 Verbal interchange of ideas; especially: CONVERSATION.

¹See <http://www.m-w.com/dictionary/discourse>.

3a Formal and orderly and usually extended expression of thought on a subject, **b**: connected speech or writing, **c**: a linguistic unit (as a conversation or a story) larger than a sentence.

4 *Obsolete*: social familiarity.

5 A mode of organizing knowledge, ideas, or experience that is rooted in language and its concrete contexts (as history or institutions) <critical discourse>.

For this thesis we will adopt the meaning of the term as stated in 2, 3b and 3c, as we view *discourse* as a *linguistic unit of connected speech or writing that is larger than a sentence*.

In general, a great variety of different types of discourse can be distinguished. What all have in common is that a speaker (by which we also mean writer) conveys information to an addressee (a term which will be used to include *listeners* and *readers*). Thus, a monologue, e. g., this thesis or any other written text, constitutes also a type of discourse where a speaker communicates with some addressees. This type of communication flow is only unidirectional as opposed to dialogic interactions where a bidirectional interaction takes place. The individual types of human-human discourse differ with respect to the number of participants as well as to the number of available modes or modalities for interaction (see table 3.1).

Type	Number of Participants	Modes	Examples
Monologue	1 – ∞	text, audio	texts, novels, self-talking, etc.
Monomodal dialogue	2	audio	telephone conversations
Multimodal dialogue	2	audio, vision, touch, scent	conversations
Monomodal multiparty discourse	3 – ∞	audio	telephone conferences, etc.
Multimodal multiparty discourse	3 – ∞	audio, vision, touch, scent	face-to-face group meetings, etc.

Table 3.1: Different types of human-human discourse and their characteristics.

An important observation is that the complexity of coordination of the interaction increases with the number of participants and available modalities. People have no difficulties to engage in a normal telephone conversation²), however, telephone conferences with more than two participants usually show frequent break-downs of the interaction protocol due to overlapping speech and confusion about who is addressed.

For human-computer interaction the different types of discourse remain basically the same. However, as table 3.2 shows, there are systems where the user only takes the role of an observer and cannot interact with the virtual agents.

²Studies in fact show that the exchange of turns in telephone conversations seems smoother than in face-to-face conversations. ten Bosch et al. (2004), for example, report on a study of spontaneous dyadic conversations. This study has shown that pauses in face-to-face dialogues last up to four times longer than pauses in telephone conversations.

Type	Number of Participants	Output Modes	Input Modes	Examples
Simulated monologue	1 virtual agent 1 – ∞ human observers	audio, vision, gestures		non-interactive performances of virtual characters
Monomodal interaction	1 virtual agent 1 human	audio, vision, gestures	gestures	interactive systems
Monomodal dialogue	1 virtual agent 1 human	audio (vision)	audio	interactive spoken dialogue systems
Multimodal dialogue	1 virtual agent 1 human	audio, visual, gestures	audio, visual, gestures	interactive multimodal dialogue systems
Simulated multiparty discourse	3 – ∞ virtual agents 1 – ∞ human observers	audio, visual, gestures		non-interactive multiparty performances of virtual agents
Multimodal Multiparty discourse	3 – ∞ virtual agents and humans	audio, vision, gestures	audio, vision, gestures	multimodal multiparty dialogue systems

Table 3.2: Different types of human-computer interaction and their characteristics.

Dyadic Discourse

Dyadic discourse or dialogue is characterized by the presence of exactly two participants that both actively contribute to the conversation. Because of the limited number of participants, dyadic conversations encompass only two conversational roles that can be taken by the participants, namely: (i) *speaker* and (ii) *addressee*. In this thesis we will use the term *speaker* to refer to the participant in an interaction that holds the speaking turn and the term *addressee* to refer to those participants to whom the speaker directs his contribution. The exact meaning of *holding the speaking turn* will be discussed later.

As we will also see later, dyadic interaction requires less coordination and synchronization efforts compared to other, more complex interaction types (see section 3.3.2). The identification of the addressee in dyadic conversations is rather trivial since the speaker usually addresses the other participant with his contribution. Most dialogues are held in order to fulfill a specific, collectively accepted goal. Common goals are, for example, the transfer of information from one party to another or the negotiation of some objective. Walton (1992) provides a classification of types and subtypes of human-human dialogue (see figure 3.1).

Multiparty Discourse

Multiparty discourse comprises, in contrast to dyadic discourse, more than two participants and thus more than the two conversational roles *speaker* and *listener*. As discussed in (Clark, 1996), the people around a multiparty conversation divide first into *participants* and *non-participants* (or overhearers; see figure 3.2). This means that all participants are ratified

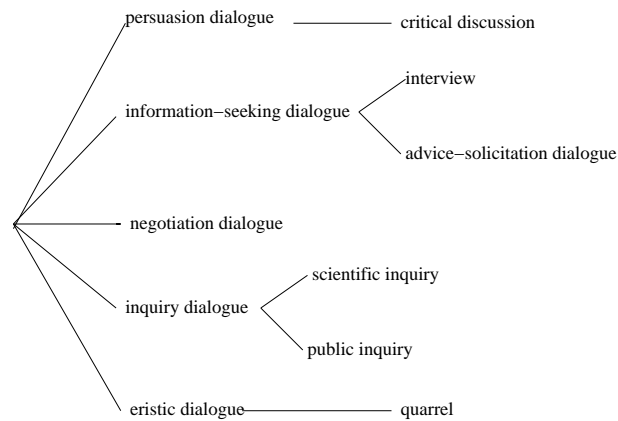


Figure 3.1: Types and subtypes of dialogue (adapted from Walton (1992)).

to take part in the interaction. Participants can take one of three conversational roles: (i) speaker, (ii) addressee(s), or (iii) side participant(s). Side participants are all people that take part in the conversation but are not currently addressed by the speaker. The other listeners are called *overhearers*. They are not allowed to participate actively in the interaction. Clark distinguishes two main classes of overhearers: (i) *bystanders* and (ii) *eavesdroppers* (Clark, 1996). While the former class is perceived by the speaker and the other participants as being present, eavesdroppers listen in on the conversation without the speaker's knowledge. The conversational role of the participants of a conversation is not fixed but rather changes

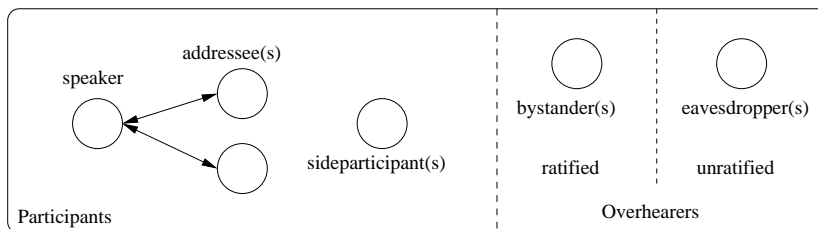


Figure 3.2: The conversational roles of participants in multiparty dialogues.

while the interaction develops. A speaker that hands over the floor to another participant immediately takes the role of an addressee or side participant, depending on whether he or she is addressed by the subsequent speaker or not. The current conversational roles of the participants and overhearers of a conversation are determined by a set of three attributes viewed from the perspective of the speaker: (i) known, (ii) ratified and (iii) addressed (see table 3.3). The actual conversational role also determines to some extent how much people understand. People who even fully overhear a conversation but do not participate (overhearer) are found to understand less than people who participate ((Schober and Clark, 1989) as quoted in (Carlett et al., 2002)). That is, at least in part, because overhearers are usually not able to alter the course of interaction, e. g., in case of a misunderstanding. However, as we will see in section 3.2.5, it is crucial for the understanding process not only to monitor the behavior

	Known	Ratified	Addressed
Addressee	+	+	+
Side Participant	+	+	-
Bystander	+	-	-
Eavesdropper	-	-	-

Table 3.3: Attributes of the different conversational roles from the perspective of the speaker (adapted from Bell (1984, p.159)).

of the other participants but also to be monitored by the others and to receive feedback from them.

Many multiparty interactions cannot function without some kind of higher level organization or structuring. Consider, for example, a small group discussion that aims to form a decision. Here, a discussion leader usually structures the contributions of the other group participants because otherwise the discussion would end in chaos. Other examples would be during business meetings, in the courtroom, in the classroom, or during talk shows on TV. Each of these gatherings has its own—more or less standardized—set of organizational rules of procedure (Cappella, 1985) and most also have a distinct participant that takes the lead and regulates the interaction. A host in a talk show on TV, for example, virtually controls the floor by selecting the next speaker after each contribution of a talk guest.

The bigger the group, the more effort needs to be spent to maintain a smooth exchange of contributions. In large groups it is difficult to monitor the individual actions of all participants which affects their behavior (see Carlett et al. (2002)). As Carlett et al. continue, in small gatherings of three participants, people tend to monitor the behavior of the other participants more carefully. As a result, speakers tend to gaze longer at someone else shortly before they are about to stop speaking. But listeners in such encounters are also more likely to employ explicit backchannel feedback, hand gestures and postural shifts—as Carlett et al. (2002) suggest, this might serve to draw the speaker’s eye. Another interesting observation is that small and rather informal groups tend to split up when about five people are participating in the interaction (see Homas (1951) as cited in Carlett et al. (2002), Dunbar et al. (1995)).

3.1.2 Social Roles and Status in Dialogues

Hulstijn (2003) distinguishes three different roles in dialogue that have different temporal scopes: (i) *dynamic roles* such as speaker or addressee that change relatively frequently, (ii) *participant roles* such as expert and novice that are stable during an interaction and (iii) *social roles* like teacher and pupil. This type of role or status is stable beyond individual interactions. In general, the status of a participant defines permissions and obligations but more importantly triggers expectations and assumptions.

The impact of roles and status on the development of interactions is best illustrated by means of teacher-pupil interactions. In the classroom, the teacher usually regulates and controls the interaction. Pupils are only supposed to speak when they are explicitly called upon. Thus, the turn-taking mechanism—which will be discussed in section 3.3.2—is at least partially invalidated. Pupils, for example, who want to take the floor, need to raise their hand in order to signal to the teacher that they want to speak. The teacher, in turn, can both accept the request and grant the turn, or decline it.

Discourse (A)	Discourse (B)
I went to the tea shop.	I went to the tea shop.
It was still open.	It was still open.
I bought some tea.	But I bought nothing.
Then I went to a supermarket.	Then I went to a supermarket.
But I bought nothing.	I bought some tea.

Table 3.4: The two discourse fragments comprise exactly the same sentences but in a different order; as a result they tell two completely different stories (adapted from Russell and Norvig (1995, p. 717)).

3.2 Propositional Aspects of Discourse

A dialogue is considered *coherent* when the individual contributions of the participants fit into the current context of the conversation. As Stenström states, “coherent utterances are also **cohesively** linked, lexically, grammatically, prosodically, and interactionally, with the immediate discourse.” (Stenström, 1994, p. 14). This means that every discourse contribution needs to be interpreted with respect to its immediate context. But more importantly, this also means that every perceived nonverbal contribution needs to be interpreted with respect to its discourse context.

We distinguish two layers of coherence: (i) coherence on the utterance level and (ii) coherence on the conceptual level. While the former is achieved by means of a high-level topical organization of the contributions, the latter is realized by means of explicit and implicit references to the external world (see section 3.2.3). However, coherence is not only a property of a discourse reflecting its topical or logical consistency. To assume a discourse to be coherent is the basis for understanding in what way ambiguous or partial utterances should be interpreted.

This section focuses on the linguistic phenomena of discourse in order to receive an understanding of the underlying motivations of the linguistic behavior of a dialogue participant. Considering the underlying motivation implies that we have to differentiate between the meaning and the communicative function of an utterance. This distinction between the meaning of the utterance and its functional aspects is related to the central concepts of the speech act theory as introduced by Austin (1962). In the following section we will consider how listeners or readers achieve a coherent representation of discourse by drawing inferences.

3.2.1 Drawing Inferences

In a technical sense, discourse or text is a conjunction of several statements where each one of them has its own meaning (Russell and Norvig, 1995). A logical interpretation of such a conjunction shows no difference between utterances $P \& Q \& R$ and $R \& P \& Q$ if the conjunction is defined as commutative. But when looking at the two discourse fragments in table 3.4, it turns out that this is certainly not true for natural languages.

Although the two discourses in table 3.4 contain exactly the same sentences albeit in a different order, it is obvious that they tell two different stories (e. g., in (A) the tea is bought in the tea shop, and in (B) it is bought in the supermarket). Because of the sequential structure of these discourses, the common interpretation is that sentences occurring later

in discourse also occur later in time. But not only the temporal order is affected by the sequence of the sentences. There are also intrinsic relations between the sentences that put the discourse together.

This means that comprehension of a discourse is not only based on the interpretation of each separate utterance within it; it is also based on a process of inference of suitable relations between the utterances. For illustration purposes, consider the following discourse fragment:

(9) *John took a train from Paris to Istanbul. His friend Peter lives there.*

Most readers will agree with the inference that the fact that John's friend Peter lives in Istanbul is the reason for John's trip to Istanbul. Even though this inference is not explicitly stated, it is natural to draw this inference under the assumption that the discourse is coherent. Generally, two different types of inferences can be identified (Eysenck and Keane, 2000):

1. **Bridging Inference**—Here the correctness of the conclusion is determined by logical principles. For example, concluding from *Peter crashed the car again* that this has happened at least once before is always correct.

Elaborative Inference—Here the correctness of the conclusion is not guaranteed. Concluding that the reason for John's trip to Istanbul is that his friend Peter lives there (9) is suitable, but not necessarily always correct. In fact he could have various reasons, there could, for example be a conference which he has to attend.

Drawing inference aims at achieving a coherent interpretation of a discourse. In general, *bridging inferences* need to be made to establish coherence between the current part of discourse and its preceding discourse. *Elaborative inferences* also account for a coherent interpretation of discourse. However, by drawing elaborative inferences, there are details added to the discourse that are not necessarily have to be correct.

3.2.2 The Structure of Discourse

The fact that listeners contemplate additional assumptions to achieve a coherent interpretation for discourse fragment (9) shows that drawing inferences is a basic process embedded in our natural language understanding capability³. As Kehler points out:

Just as we attempt to identify syntactic and semantic relationships when presented with a sequence of words in an utterance, we attempt to identify coherence relationships when presented with a sequence of utterances in a discourse. (Kehler, 2002, p. 3)

This leads to the widely accepted notion that a set of implicit relations between the sentences of discourse exists, so that the content of one sentence might provide justification, elaboration or explanation for the content of another. Two different kinds of occurrences of these relations are observable:

- 1 Sometimes speakers make such relations by explicitly using particular *cue phrases*, linguistic elements like “*but*” or “*for example*.”

³Parts of this section are based on material from Pflieger (2002).

forward a theory of *coherence relations* based on a limited, but bigger, set of coherence relations, applied recursively to discourse segments. However, since Hobbs focuses his research on textual discourse at the sentential level, his approach is not suitable for the phenomena of greater parts of discourse, like paragraphs.

Mann and Thompson put forward the *Rhetorical Structure Theory* (RST) (Mann and Thompson, 1987), in which they introduce so-called *rhetorical relations*. One major claim within this approach is that the same rhetorical relations that hold between individual sentences also account for the relationships between larger segments of discourse. These *rhetorical relations* resemble the *coherence relations* of Hobbs and additionally take into account which effect should be caused with the reader. Most *rhetorical relations* are asymmetric. One element of the pair of connected discourse entities is central (the *nucleus*) and the other part is peripheral (the *satellite*) depending on their respective contribution to the meaning of the next higher level. Although it is not a “computational” theory, Mann’s and Thompson’s theory is one of the most influential ones in this area.

Dialogue Structure

Until now we have considered coherence of discourse in general. We now want to particularly address the coherence of dialogues. Similar linguistic phenomena like referential relations, elliptic expressions, and coherence relations account for the coherence between turns (moves) of the (at least two) participants of a dialogue. Since a dialogue is an interactive attempt to achieve an overall goal, the question of whether an utterance fits in a coherent manner into its preceding dialogue context is based on the decision whether and in which way it contributes to the achievement of this overall goal (the task of the dialogue). What is needed is a segmentation of the dialogue into *discourse segments* yielding to group utterances sharing a common subject, viewpoint, or purpose. These discourse segments correspond to subgoals of the overall goal of the dialogue.

One interesting computational approach to modeling the structure of dialogue is to map the utterances to a hierarchical set of *dialogue acts*. As it is pointed out in (Bunt, 2000), the meaning of an utterance can be viewed in terms of context changes by distinguishing between the information the speaker is introducing into the context (*semantic content*) and the way information has to be inserted into the context in order to play the intended role (*communicative function*). The combination of semantic content and communicative function of an utterance is called *dialogue act* (see section 2.2.4). Dialogue acts can be used to describe the meaning of an utterance. Bunt defines the meaning of dialogue acts in terms of changes in a context representation.

However, this implies that the dialogue follows a pre-determined plan, which is the case in task-oriented dialogues but not necessarily in information-seeking dialogues where the user might change their mind and switch between different applications depending on the response of the system. Ahrenberg et al. introduce small-range discourse plans called *initiative-response units* (*IR-units*) which consist of an *initiative* (a turn opening the segment) and an appropriate *response*—a turn closing the segment (Ahrenberg et al., 1991). The response does not have to be adjacent to the initiative, however, it has to follow at some point in time. In the meantime, a clarification sub-dialogue or a preconditioned dialogue for the requested action can take place. In Ahrenberg et al. (1991) the IR-units listed in table 3.5 are distinguished.

In order to describe the structure of interactions, Stenström defines five hierarchical levels

IR-unit	Type of initiative	Type of Closing Response
Q/A	Information Request	Answer
Q/AS	Information Request	Assertion that implies failure to answer
D/ACK	Directive	Acknowledgment (after execution)
AS/ACK	Assertion	Acknowledgment (of information)
AS/AS	Assertion (of interest, want)	Assertion (forwarding interest)

Table 3.5: Examples of Initiative-Response units (Ahrenberg et al., 1991).

The TRANSACTION	consists of one or more exchanges dealing with one single topic; one or more transactions make up a conversation
The EXCHANGE	is the smallest interactive unit consisting, minimally, of two turns produced by two different speakers
The TURN	is everything the current speaker says before the next speaker takes over; it consists of one or more moves
The MOVE	is what the speaker does in a turn in order to start, carry on and finish an exchange, i. e., the way s(he) interacts; it consists of one or more acts
The ACT	signals what the speaker intends, what s(he) wants to communicate; it is the smallest interactive unit

Table 3.6: The five hierarchical levels of spoken interactions defined by Stenström (1994, p. 30).

that serve to describe spoken interactions (see table 3.6). Each level of this hierarchy consists of one or more units from the level next below. The lowest level of this hierarchy deals with the propositional content of the individual utterances and will be discussed in this section.

Contextual Enrichment

Every contribution has to be interpreted with respect to its context of use. However, besides the interpretation of referring expressions, there are other situations where contextual information contributes to the meaning of an utterance. For human-computer interactions, Chai and Jin (2004) discuss, for example, so-called *context questions* or follow-up questions as in the following example (taken from the SMARTWEB scenario):

- (11) Q1: “*Who won the World Cup in 2006?*”
 Q2: “*How often did this team win a World Cup?*”

In this example, the interpretation of Q2 depends on Q1 since the referring expression *this team* can only be resolved in the context of Q1 (see section 3.2.3). Chai and Jin (2004) identified three types of discourse transitions for the contextual interpretation of contributions⁴:

⁴Note that Chai and Jin (2004) focus on question-answering dialogues and thus only consider questions. But most of their results can be generalized to other kinds of discourse.

- 1 *Topic Extension*: A contribution that concerns a similar topic but with different constraints, as in the following example (taken from the SMARTWEB scenario):

(12) Q6: “What’s the weather forecast?”

Q7: “What’s it for tomorrow?”

- 2 *Topic Exploration*: A contribution that concerns the same topic as the previous utterance but focuses on a different aspect, as in the following example from the SMARTWEB scenario:

(13) Q8: “Who won this year’s FIFA World Cup?”

Q9: “Who scored the goal?”

- 3 *Topic Shift*: A sequence of contributions that concern two distinct topics.

(14) Q10: “What is the name of the volcano that destroyed Pompeii?”

Q11: “How many people were killed?”

Thus, given two subsequent questions (Q_i and Q_{i+1}) information can be transferred from Q_i to Q_{i+1} . As we will see in the next section, this type of transfer of contextual information can be explained by the discourse phenomenon of bridging references (see below).

3.2.3 Referring Expressions

A key linguistic phenomenon of verbal interaction is the use of referring expressions like *Michael Ballack* or *he* to denote or refer to a person named Michael Ballack, as in the following example (taken from the SMARTWEB scenario):

(15) **User**: “Which team is Michael Ballack playing for?”

System: ...

User: “How often did he score?”

In this discourse fragment, *Michael Ballack* and *he* are both referring expressions that refer to a person named Michael Ballack (their referent). Thus, a *referring expression* is a natural language expression used to perform a reference, and the entity of the real world that is referred to is called the *referent*. If two referring expressions denote the same entity, they are said to *corefer*. A referring expression that refers to some entity or concept of the physical, situational or discourse context is called *deixis* or *deictic expression*. Examples of deictic expressions are personal pronouns (e.g., *I*, *you*), adverbial expressions (e.g., *here*, *now*) and demonstrative pronouns (*this*, *that*). However, the category *deixis* also subsumes referring expressions relating to an entity that has been introduced during previous discourse.

A general characteristic of deictic expressions is their immanent context dependency. Thus, referring expressions can only be interpreted with access to their context of use. Another aspect emphasizing the role of referring expressions is their contribution to efficient and coherent communication. Grice’s maxims of quantity and manner (Grice, 1975), for example, postulate that speakers should keep their contributions brief and as informative as necessary and this can only be achieved by using referring expressions.

But before we will consider the characteristics of referring expressions in detail, we will first discuss how referring expressions account for the coherence in discourse (see section 3.2.1).

Type	Example
Direct reference	
Identity:	<i>I met a man yesterday. The man told me a story.</i>
Pronominalization:	<i>I met a man yesterday. He told me a story.</i>
Epithets:	<i>I met a man yesterday. The bastard stole all my money.</i>
Set membership:	<i>I met two people yesterday. The woman told me a story.</i>
Indirect reference by association	
Necessary parts:	<i>I looked into the room. The ceiling was very high.</i>
Probable parts:	<i>I walked into the room. The windows looked out the bay.</i>
Inducible parts:	<i>I walked into the room. The chandeliers sparkled brightly.</i>
Indirect reference by characterization	
Necessary roles:	<i>John was murdered yesterday. The murderer got away.</i>
Optional roles:	<i>John died yesterday. The murderer got away.</i>
Reasons, causes, consequences and concurrences	
Reasons:	<i>John died yesterday. The murderer got away.</i>
Causes:	<i>John fell. What he did was trip on a rock.</i>
Consequences:	<i>John fell. What he did was break his arm.</i>
Concurrences:	<i>John is a Republican. Mary is not so smart either.</i>

Table 3.7: Taxonomy of bridging references as they occur in natural discourse (adapted from Clark (1977, pp. 170-172)).

Bridging References

Clark (1977) introduces a taxonomy of bridging references that he observed to occur naturally in discourse. He considers bridging an obligatory part of the process of comprehension that is based on the distinction between given and new information. During the course of interactions, speakers link the given information to previously mentioned information or knowledge the listeners are supposed to know. This, however, can only work when the speaker and the listener enter an agreement that Clark and Haviland termed the *Given-New Contract* (see Clark and Haviland (1974)):

The speaker agrees to try to construct the Given and New information of each utterance in context (a) so that the listener is able to compute from memory the unique Antecedent that was intended for the Given information, and (b) so that he will not already have the New information attached to the Antecedent. (Clark, 1977, p. 170)

Bridging from previous knowledge to the intended antecedent can take on many forms so that any taxonomy can hardly be complete. However, Clark's taxonomy covers the most important variants of bridging and illustrates the difference between direct reference (i. e., anaphora) and implicit references (see table 3.7).

The first type of bridging reference—the direct references—is usually covered by the reference resolution components of state-of-the-art dialogue systems, the second and third type, however, are usually not supported since their resolution requires a great deal of world knowledge and inference. In Chapter 8 we will show how the second type—the implicit or associative

references—can be processed via access to what in our implementation is called a long-term memory.

In the following, we will give an overview of the different types of referring expressions as they occur in spoken dialogues. The interpretation of referring expressions has been extensively studied in theoretical linguistics, computational linguistics and psycholinguistics. We will concentrate on a subset of this wide area and focus on the interpretation of anaphoric expressions based on the discourse context (see the next section *Anaphora*), the interpretation of referring expressions whose referents are located in the physical or situational context (see section *Deixis*).

Anaphora

One of the simplest forms of *bridging reference* (see the previous section) is involved in the interpretation of *anaphora* (from the Greek word *anaphorá*, “carrying back”), where a pronoun or nominal phrase (NP) has to be linked with the previous mentioning of a noun or NP. Generally speaking, anaphora is a reference to an entity in the preceding discourse in the expectation that the receiver will be able to identify the referenced entity.

Various definitions of anaphora have been suggested, one classic definition is given by Halliday and Hasan. It is based on the notion of cohesion: “*anaphora is cohesion (presupposition) which points back to some previous item*” (Halliday and Hasan, 1985). According to this definition we have to distinguish between anaphora and *cataphora* which occurs when the anaphora precedes the antecedent, for example:

(16) *Since she was very hungry, Mary was looking for a restaurant.*

The interpretation of anaphoric expressions (called *anaphora resolution*) requires referential support from the surrounding context. An anaphora must be linked to the element of the prior discourse to which it is “pointing,” the *antecedent*. Both the anaphora and the antecedent are used as referring expressions and in the case of a total anaphora (see below) refer to the same *referent* in the *real world*. Luperfoy (1991) points out that the linkage between the anaphora and its referent is established by an asymmetric *sponsoring* relation. Furthermore, Luperfoy stresses: “More importantly [...] the sponsorship relations accommodate the full range of possible dependency behavior which can be obtained between referent and dependent, including, but not limited to coreference.” (Luperfoy, 1991, p. 9). When we now consider the total/partial anaphora dichotomy, the need for such a sponsorship relation will become obvious.

Total Anaphora versus Partial Anaphora

The distinction between total and partial anaphora shows how an anaphora relates to its referent. *Total anaphora* is the simpler condition where a pronominal NP co-refers with its antecedent. An example for total anaphora is given in sentence (17), *he* and *it* both being examples for total anaphora. *He* is sponsored by *Peter* and shares the meaning with its antecedent entirely, whereas *it* is sponsored by *the car*.

(17) *Peter stopped the car. He thought it had a flat tire.*

A *partial anaphora* is the result of the referring expression (e. g., a NP) not co-referring with its antecedent. In this case, the referring expression only incorporates part of the meaning

from its referent and contributes the rest itself. In (18) the two definite NPs *the pilot* and *the window* do not refer to the same entities as their antecedent does:

- (18) *Peter saw a plane. The pilot was waving at the window.*

Luperfoy (1991) points out that it is the prior mentioning of a plane that refers to the mentioning of a pilot. The denotation of the dependent NP is an extended part of the denotation of the antecedent. Thus, partial anaphoric references realize indirect references in the sense of Clark (1977). The partially anaphoric use of full lexical definite NPs is constrained by the contrast between familiarity and novelty of the definite NP. Consider for example these sentences:

- (19) *Peter saw a plane. The old jet fighter looked unfit to fly.*

The definite NP *the pilot* in example (18) is novel to the discourse whereas the definite NP *the old jet fighter* in example (19) is familiar to the discourse and therefore refers to exactly the same plane the antecedent NP *a plane* did.

But of course not all pronouns in English are anaphoric. The pronoun *it* can also be used in a non-anaphoric way (e. g., *It is raining*). Another interesting observation is that people, when interacting with dialogue systems, tend to use definite descriptions for references to the visual context without introducing the antecedent by means of an indefinite NP. With respect to multimodal dialogue systems, Boye et al. (2004) confirm this observation by means of two independent corpora of Wizard-of-Oz experiments with two multimodal dialogue systems (the ADAPT system (Gustafson et al., 2000) and the NICE system (Gustafson et al., 2004)).

Exophora

Exophora (Greek: *exo*—outside; *pherein*—carry) subsume all references to extra linguistic entities, i. e., entities that can be found in the situational or visual context of the discourse. This kind of reference is also called *cross-modal* reference. Exophora can be expressed through deictic references (e. g., *this*, *that*; see below), by *Homophoric references*⁵, but also by means of definite nominal phrases to refer to objects in the field of view, as in the following examples (the first example is taken from the VIRTUALHUMAN scenario and the second one from the OMDIP scenario):

- (20) *“Move Ballack into the central midfield”* [In the context of a list of football players displayed on the screen.]
- (21) *“Play the third song.”* [In the context of a list of songs displayed on the screen.]

What these two examples have in common is that the actual antecedents of the referring expressions were never explicitly introduced into the discourse. However, the antecedents can be found in the visual or physical context of the interaction. What is also important is the fact that speakers tend to incorporate disambiguating information into their referring expressions if more than one potential referent of that kind exists. Salmon-Alt introduced the term *differentiation criterion* for this disambiguating information (Salmon-Alt, 2000). Examples of differentiation criteria would be color or shape.

⁵In contrast to deictic expressions, homophoric references depend on socio-cultural knowledge and context, rather than on specific features of a particular physical or verbal context. Consider, for example, phrases like *“the president”* which denotes different entities when it is used in the U.S. or in France or *“the cat”* which denotes a particular cat when it is used by the owner of the cat.

Visual Perception and Focus Even though virtually all entities that are available in the field of view of the participants can serve as potential antecedents, not all of them are equally accessible (see, for example, Maass (1996); Byron et al. (2005)). Byron et al. (2005) discuss a computational model of visual attention that is based on several factors that influence the focus of our visual attention when presented with complex scenes.

The first factor is based on the *preattentive visual processing stage*, a stage that is known to be sensitive to features like color, orientation or size. Objects that appear to be novel with respect to this feature space are in the focus of attention during later processing stages. The second factor of Byron et al. is related to two time spans that define the salience of an object if an object falls out of view: (i) the amount of time since it was last seen (*recall delay*) and (ii) the exposure time before the object dropped from view. Based on these factors it is possible to determine the salience of visually perceived objects.

Deictic Expressions

Deixis (Greek: *display, demonstration, or reference*) is a form of reference that absolutely relies on its context of use. Deictic expressions are usually encoded by means of demonstrative pronouns, personal pronouns in the first and second person, temporal, tenses or specific adverbs that have some temporal or spatial impact, like *here* and *now*. In order to be able to understand a deictic expression, a listener of a deictic expression must have access to this context, as the following example from the OMDIP system shows:

(22) **A:** “Play this [pointing gesture] song.”

Here, the pronoun *this song* does not refer to a particular song (as when it is used in an anaphoric way) but rather it is a place-holder or a variable for an entity that is determined by the context (i. e., a pointing gesture). As Levinson (1983) highlights, this particular factor reminds us of the usage of speech that is normally planned for face-to-face interactions.

Usually, deictic expressions are organized in an egocentric way around a *deictic center* that defines the following anchor points, which are typically unmarked (Levinson (1983)):

- 1 The speaker is the central person.
- 2 The central time is the moment when the expression is uttered.
- 3 The central location is the spatial location of the speaker during the production of the utterance or coding time.
- 4 The discourse center is the location of the speaker during the production of the utterance.
- 5 The social center is the social status of the speaker during the production of the utterance.

It is also important to distinguish between the different types of usage of deictic expressions: gestural and symbolic usage (Fillmore (1971) as quoted in Levinson (1983)). Deictic expressions that are used in the gestural condition can only be resolved by having access to the physical context. As Levinson (1983) put it, one would need at least a video recording to be able to resolve gestural deictic expressions. This is for example the case when the speaker uses pronouns in the third person that are accompanied by gaze gestures as in:

(23) **A:** “*You two, come with me!*” [gazes at participant **B** and **C**]

This means that a listener of a gestural deictic expression needs to monitor the physical environment on a moment-by-moment basis. Symbolic deictic expressions, however, can be resolved without access to the physical context. It is sufficient to have some knowledge of the basic spatial context of an utterance, as in the following example from the SMARTWEB system:

(24) **User:** “*How do I get from here to Berlin?*”

In this example, the system only needs to know where the speaker is located in order to understand the complete utterance. Thus, symbolic deictic expressions can be resolved by means of a broader situational context than gestural ones.

In general, deictic expressions can be categorized into four classes which we will discuss in the remainder of this section: (i) *person deixis*, (ii) *time deixis*, (iii) *place/spatial deixis* and (v) *discourse deixis*. However, there are other categories that are out of scope for this thesis such as:

Social deixis which concerns the encoding of social differences among the participants and the referenced entities.

Empathetic deixis which can be used by the speaker to indicate their emotional closeness or distance from the referent.

Person Deixis Person deixis concerns the encoding of the role of the participants of an utterance. The basic grammatical differentiation is realized through the three grammatical categories of person. The first person is used by speakers as a reference to themselves—the speaker uses the singular *I* (German *ich*) as a reference to themselves and the plural *we* (German *wir*) as a reference to a group of people that includes the speaker. The second person is used to reference one or more addressees (English *you*, German distinguishes between the singular *du/Sie* and the plural *ihr/sie*). The third person is used to reference people that are neither the speaker nor the addressee of the utterance. Like German there are several languages where the first, second and third person are all marked for singular and plural forms.

What is important for the interpretation of person deixis, is that when the speaker changes, the deictic center is also moved from the previous speaker to the current one, as in the following example:

(25) **A:** “*You hang up first!*”
B: “*No, you hang up first!*”

Moreover, participants that perceive such an utterance must be able to identify whether they are addressed or not in order to completely understand the utterance; a process that is not that easy for multiparty conversations (see section 3.3.3). Things get more complicated when the communication is transmitted by another person, e. g., a messenger who delivers a verbatim message from another speaker. This means that the speaker and the source of the message are not the same and this could cause confusion. Levinson (1983) outlines an independent pragmatic model of roles that participants can take in a conversation. In this model he distinguishes between the *utterance-source* and the *speaker* and between the *receiver* and the *target*.

Time Deixis Time deixis concerns the encoding of time points and time spans relative to the time point when the utterance is produced. This is referred to as the coding time (CT) of an utterance, which can be different from the receiving time (RT) (Fillmore (1971) as quoted in Levinson (1983)). Time deixis is usually expressed through deictic temporal adverbs like *now*, *then*, *yesterday* or *next year*. However, temporal expressions can also be used in a calendric way which means that they refer absolutely to some kind of *origio* or starting point of time.

What makes the interpretation and resolution of temporal deixis even more complex is the ambiguity between their relative and absolute interpretation when we consider complex compound temporal reference like *next Monday*, *next year*, or *this week*. These ambiguities appear in particular in complex temporal expressions that consist of a deictic reference (such as *next*, *this*, *last*) that modifies a non-deictic name (such as *Monday*, *December*) or unit of time (such as *week*, *month*, *year*).

Moreover, there is also some kind of a cultural convention for the interpretation of absolute reference to week days. Consider, for example, the following temporal deixis (taken from the SMARTWEB scenario):

(26) “*What’s the weather forecast for Saturday?*”

A receiver of this message will come up with different interpretations depending on what day the message is uttered. If the message is uttered on a Saturday the listener would infer that the speaker meant *next Saturday* (one week later). Thus, if the speaker means *today*, she or he must use this term instead of the name of the weekday. Even if the message is uttered on Friday, the listener would infer that the speaker meant *next Saturday*. In this case the speaker should have used *tomorrow* in order to achieve the intended interpretation. In general, there is a strong preference for the use of deictic references (like *yesterday*, *today*, *tomorrow*) over the use of calendric or absolute references (Levinson, 1983).

Some temporal expressions express only vague temporal reference and it is often difficult to precisely resolve these references. Expressions such as “*in several weeks*,” “*tonight*” or “*by Monday at the latest*” are difficult to represent by points or exact intervals in time. As we have shown in the context of the SMARTKOM system, at least some domain knowledge about the task at hand is needed in order to be able to resolve these references. For instance, if someone wants to reserve a table in a restaurant, *tonight* would mean between 7 and 9 pm. However, this mapping is again subject to cultural differences. Germans, for example, are used to going to a restaurant between 7 and 9 pm while Spanish people (at least those from the south) would never go out before 10 pm.

Place/Spatial Deixis Place or spatial deixis is used to encode spatial references relative to the current location of the participants of an utterance or relative to other objects mentioned during the discourse. Most languages provide means to distinguish at least between *proximal* and *distal* entities (e. g., adverbs like *here* vs. *there* or demonstrative pronouns like *this* vs. *that*).

In general, the location of an object can be uttered in two different ways, either by describing its location relative to the location of other objects (see example 27; taken from the SMARTWEB scenario) or by using deictic expressions that are related to the current location of the participants at the coding time (see example 28; taken from the SMARTWEB scenario).

(27) “What’s the name of the player to the right of Ronaldo?”

(28) “What’s the name of the second player from the right?”

In the first example, the referenced player is referenced relative to another player (here, Ronaldo) who the addressee is supposed to know or can at least easily identify. In the second example, the player is described with respect to a specific location by means of the deictic expression “*from the right*.” However, as Levinson (1983) stresses, it is important to consider the pragmatic meaning of place deixis since there is a big difference in the precision of the action in example (29) depending on whether we ask a surgeon or a bulldozer driver.

(29) “Place it **here**.”

The interpretation of these expressions depends on the point-of-view the speaker adopts, which is called the *frame of reference*. As Levinson (2003) states, the notion of frames of reference is crucial for the study of spatial language. The phrase “frame of reference” originates from Gestalt theories of perception in the 1920s. In general, frames of reference come down to the selection of reference objects, for instance, when someone walks down the street while talking on a cell-phone to someone else. What happens to the cell-phone; is it moving? It depends on the frame of reference: the speaker or the street.

The structure and distinctions between frames of reference have been a research topic in various disciplines such as philosophy, cognitive science, psychology and linguistics. Of course this led to various conceptions. Based on a detailed review of these different notions of frames of reference, Levinson distinguishes three main frames of reference: *intrinsic*, *relative* and *absolute* (Levinson, 2003, p. 35). Frames of reference are typically described by a set of primitives like:

- **Labeled angles**—language specific labels, e.g., *left*, *front*, *south*.
- **Figure (F)**—the object to be located.
- **Ground (G) or relatum**—the object with respect to which the figure is to be located.
- **Viewpoint (V)**—the viewpoint of the observer.
- **Anchor point (A)**—to fix labeled coordinates.

Following Levinson (2003), we give a brief characterization of the three types of frame of reference in table 3.8.

Vagueness of Spatial Expressions The interpretation of spatial references is not always unambiguous. In fact, it is often not possible to determine whether such a referring expression has a deictic or non-deictic interpretation. Consider, for example, the following utterances:

(30) “The cat is behind the car.”

(31) “Could you hand me the left book?”

In example (30), it is not clear whether *behind* should be interpreted as deictic expressions (the cat is between the car and the speaker) or a non-deictic expression (the cat is at the rear of the car). Nevertheless, this ambiguity causes only minor confusion in human interactions,

<i>Intrinsic frame of reference</i>	When using an intrinsic frame of reference, the speaker takes the viewpoint of the relatum. Thus, this frame of reference involves an object-centered coordinate system. The location of the figure is determined by means of aspects of the relatum like sidedness of facts.
<i>Relative frame of reference</i>	When taking a relative frame of reference, the speaker locates an object relative to his or her own viewpoint and another object. Thus, the speaker spans a triangulation of three points: the viewpoint V, the ground G and the figure F.
<i>Absolute frame of reference</i>	The absolute frame of reference is characterized by the fact that an unambiguous reference point is adopted, i. e., the four cardinal points.

Table 3.8: Characterization of the three main frames of reference (Levinson, 2003).

as speakers tend to either add additional disambiguating information (e. g., “*Could you hand me the left book? I mean from my point of view.*”), or stick to the frame of reference that was selected in a previous turn. The second option is of particular interest for our work, as it shows the necessity to keep track of the selected frame of reference during the course of an interaction.

Pars-pro-toto Deixis When referencing objects that are displayed on the screen of a multimodal dialogue system through pointing gestures, people sometimes tend to reference only parts of the object they actually intend to reference. Wahlster defined this type of pointing gestures as *pars-pro-toto* deixis which is a subtype of place deixis (Wahlster, 1991). Following Wahlster (1991), this type of pointing gesture can occur in two different ways: (i) a pointing gesture that selects an embedded object of the actual referent, or (ii) a pointing gesture selecting an arbitrary part of the actual referent.

Discourse Deixis In addition to the anaphoric referring expressions discussed in the previous sections, there are referring expressions that have no NP-antecedent. These anaphora are *discourse deictic* since they refer to verbal complexes, clauses, whole utterances or text sections of the previous context (Webber, 1991; Levinson, 1983). Eckert (1998) shows with the following examples that discourse-deictic reference can occur on different levels:

- (32) A: “*John crashed the car.*”
 B: “**That** happened yesterday.”
- (33) A: “*John crashed the car.*”
 B: “**That**’s not true.”
- (34) A: “*John crashed the car.*”
 B: “**That**’s surprising.”
- (35) A: “*My number is 3 4 7 2 4.*”
 B: “*I’m sorry, could you repeat that?*”

The anaphora can refer to the event (32), the proposition (33), the fact expressed by the previous clause (34) or to the utterance itself (35). In English, ambiguities between NP- and discourse-deictic reference can occur. Consider, for example, the following two discourse fragments:

- (36) **A:** *“Peter bought a bike.”*
B: *“It’s great.”*
- (37) **A:** *“Peter bought a bike.”*
B: *“That’s great.”*

Both discourse fragments have two readings, however, there is a strong preference to interpret pronouns as NP reference as in (36) and demonstrative as discourse-deictic reference as in (37). In general, discourse deictic references are very common in everyday conversations and are often used to express one’s opinion about past utterances of other participants or to express agreement. Consider, for example, the following discourse fragments which are both taken from the VIRTUALHUMAN system:

- (38) **A:** *“I would suggest to putting Ballack into the central midfield.”*
B: *“Let’s do that.”*
- (39) **A:** *“Answer B is correct.”*
B: *“Yes, I agree.”*

As (39) shows there are situations where a discourse reference occurs without an explicit referring expression. Nevertheless, these examples show that discourse deixis in spontaneous speech basically works in the same way as in written discourse. However, if more than two people participate in a conversation, things get more complicated since it is possible to refer back to the last utterance of each participant:

- (40) **Peter:** *“Answer B is correct.”*
Maria: *“No, answer C is correct.”*
Paul: *“Peter is right.”*

Syntactic and Semantic Constraints on Reference

Resolving referring expressions means finding a matching referent and establishing a linkage between the referring expression and this referent. The actual search for the matching referent is restricted by a set of constraints. Among these constraints are syntactic and semantic restrictions that are defined by the referring expressions themselves and their role in the sentence (i. e., the selectional restrictions that a verb places on its arguments).

Syntactic Constraints Referring expressions, usually, must agree in number, person and gender with their referents. The following examples (adapted from Jurafsky and Martin (2000)) illustrate constraints on number and person agreement for English:

- (41) *John has a new car. It is red.*
(42) * *John has a new car. They are red.*

and for German:

(43) *Peter besitzt ein neues Auto. Es ist rot.*

(44) * *Peter besitzt ein neues Auto. Sie sind rot.*

Besides number agreement, the referring expression and its referent must also agree in gender and case. English and German third pronouns distinguish between male, female and neuter (nonpersonal). In English the first two only apply to animate entities while German also allows for personal reference to inanimate objects. The following examples illustrate the gender and case agreement:

(45) *John and Mary have new cars. They love them.*

(46) *Marie und ich haben neue Nachbarn. Morgen treffen wir uns.*

In (45) the pronoun *they* refers to *John and Mary* and *them* refers to *cars*. This is the only way to resolve these two referring expressions since *John and Mary* and *they* are in the nominative case and *cars* and *them* are in the accusative case. The same holds for the German example (46), the referring expression *wir* is in the nominative case and can only refer to the nominative of the previous sentence while *uns* is in the accusative case and thus can only refer to the accusative of the previous sentence.

Semantic Constraints Besides syntactic constraints, the semantic role of a referring expressions also helps to determine the correct antecedent. Consider, for instance, the following example that we adapted from Jurafsky and Martin (2000):

(47) *John parked his car in the garage. He had driven it around for hours.*

In this example there are two potential referents for the pronoun *it*, the car and the garage. However, the verb *drives* puts some constraints on its direct object, i. e., the object must be something that can be driven like a truck or a car. A garage, in contrast, is a static object that cannot be moved so that *car* is the only referent that fulfills this semantic constraint. As Jurafsky and Martin (2000) note, there are more general semantic constraints that are much more difficult to encode in a comprehensive manner. Consider the following two passages (adapted from Jurafsky and Martin (2000)):

(48) *John parked his car in the garage. It is incredibly messy, with old bike and car parts lying around everywhere.*

(49) *John parked his car in the garage. It is incredibly messy, with old bike and car parts lying around in the trunk.*

(50) *John parked his car in Beverly Hills. It is incredibly messy, with old bike and car parts lying around everywhere.*

In passage (48), the pronoun *it* is most probably the *garage* since the parts mentioned in the paragraph are lying around *everywhere*. In contrast, in passage (49) the *car* is almost certainly the referent for *it* as in this case the parts are lying around in the trunk which is part of a car and not part of a garage. The last passage (50), however, shows that world knowledge also influences the process of resolving the pronoun *it*. Anybody who knows Beverly Hills would most probably resolve *it* with the *car*. Thus, semantic constraints require a lot of world knowledge and intuition and this is not always easy to encode in practical algorithms.

Accessibility of Noun-Phrase-Antecedents

Not all concepts that are mentioned during discourse are equally accessible for producing and understanding referring expressions. Moreover, speakers can select what kind of referring expressions they will use out of a number of possibilities. When referring to a particular car, for example, one might use *it*, *this*, *that*, *this car*, *that car*, or *the car* to encode the reference. However, as Kehler (2000) stresses, these alternatives are not freely interchangeable. Each encodes a slightly different signal about where the listener will find its referent in their mental model of the discourse—the referent’s *cognitive status* (see Gundel et al. (1993)) .

Gundel et al. (1993) presented a theory of cognitive status based on a *Givenness Hierarchy*. This hierarchy defines six cognitive statuses that referents can have and the types of referential expressions that signal them (see table 3.9). These six statuses are *inclusive*, i. e., each status

Referential Accessibility	Example
in focus	<i>it</i>
activated	<i>that</i>
familiar	<i>that car</i>
uniquely identifiable	<i>the car</i>
referential	<i>this car</i>
type identifiable	<i>a car</i>

Table 3.9: Givenness hierarchy for noun-phrases (Gundel et al., 1993).

implicates all subordinate statuses (if a referent is considered *activated* this means it is also *familiar*). This is in-line with the assumed structure of the human memory. Starting with *familiar*, there is a representation of the concept in memory. Every concept of status *activated* is located in the short term memory. A concept that is newly introduced into the discourse and that cannot be linked to a referent is of status *type identifiable*. Listeners only need to know the meaning of the NP. Indirect references (*Inferrables*) typically receive the status *uniquely identifiable* but not *familiar*.

Summary

In this section we have discussed the central aspects of referring expressions and have given an overview of the types that are distinguished in the literature. However, until now we have taken for granted the idea that it is possible to unambiguously identify the type of referring expressions. This unambiguous classification of referring expressions would make it easy to take the appropriate steps for reference resolution. The examples of the different types of referring expressions have shown that some types of references can be identified without access to the discourse context while others are ambiguous, i. e., they require some more elaborated interpretation. Consider, for example, the following utterance (taken from the OMDIP scenario):

(51) **User:** “Play this song!”

Here, the reference could either be a place deixis if accompanied by a pointing gesture or an anaphoric reference to a previously mentioned song. The point is, it is not clear until the process of reference resolution is finished and one would need to test several hypotheses

before the type can be identified unambiguously. In section 8.6.2 we will pick up this aspect again and present a hierarchical set of hypotheses that guide the resolution process.

3.2.4 Ellipsis and Fragments

This section gives a short overview of the research in the area of VP-ellipsis constructions.⁶ First we will discuss in short textual VP-ellipsis and then we will turn to a special case of elliptical constructions in dialogues (see section 3.2.4) In this overview we consider only elliptical constructions whose interpretation make use of resources or processes associated with the semantic aspect of discourse. The gapping construction, for example, which is characterized by an elision of all but two bare constituents, is considered to be based on syntactic phenomena. Since our approach focuses on the processes associated with discourse context, syntactic phenomena like the gapping construction will not be treated here. Characteristic for VP-ellipsis in English is the stranded auxiliary verb which indicates the elision of a verb phrase, exemplified by this sentence:

(52) *Peter likes Maria, and Martin does too.*

The interpretation of a VP-ellipsis requires that the meaning of the missing VP has to be recovered. This is achieved by incorporating the meaning of the VP of an appropriate antecedent clause. In (52) this would be the first clause. The intended meaning of this VP-ellipsis is that *Peter* also likes *Maria*. The VP-ellipsis in (52) has only one reading but ambiguities can emerge when the antecedent clause contains a pronoun or other context-dependent forms, like in the following example:

(53) *Peter likes his girlfriend, and Martin does too.*

This sentence can be interpreted in two different ways. First it could mean that *Martin* likes *Peter's* girlfriend. It could also mean that *Martin* likes *his own* girlfriend. These interpretations are called *strict* and *sloppy* readings, respectively. In German, the ellipsis construction is characterized by the elision of identical material. Consider, for example, the following sentence (taken from Bußmann (2002)) which shows an elision of the verb *trank* in the target clause:

(54) *Er trank Bier und sie Wein.*
(*He drank beer and she (drank) wine.*)

There is an ongoing debate over the level of language processing on which a VP-ellipsis is resolved. In the literature two contrasting positions can be identified. The first position is based on the assumption that a VP-ellipsis is resolved on the syntactic level (see for example Lappin (1996)), whereas the second position is based on the assumption that it is resolved at a purely semantic level of representation (see for example Kehler (1993) or Hardt (1999)). The syntactic approaches posit that syntactic material is copied from the antecedent clause and integrated in the elided clause, whereas the semantic approaches posit that this material is retrieved from the semantic representation of the antecedent clause.

For both approaches there are large sets of data offering evidence to support their particular view. This is achieved by predicting the unacceptability of elliptical constructions which

⁶For an exhaustive treatment of VP-ellipsis constructions in written discourse, the reader is referred to Hardt (1999) and Kehler (2002).

the contrastive approach would predict as acceptable. Kehler (2002) presents data demonstrating in a systematic way that the coherence relation operating between the source and target clause is an indicator at which level the VP-ellipsis has to be resolved. Kehler claims that three classes of *connections between utterances* exist, which are *resemblance*, *cause-effect* and *continuity*. In particular, the predictions of the syntactic approaches fit the data when a resemblance relation occurs and the predictions of the semantic approaches fit the data when a cause-effect relation occurs.

Carbonell (1983) reports on an empirical study investigating how humans interact with natural language interfaces. The emphasis of this study was placed on the use of several linguistic phenomena including elliptical constructions. The significant result was that users appear unwilling to construct complex commands, queries or statements when they can express the same proposition in a more succinct manner. Given these results, Carbonell concluded that it is more appropriate to handle abbreviatory discourse devices. One of these abbreviatory discourse devices is a special form of ellipsis which we will consider in the following subsection.

Elliptical Constructions in Dialogues

There is another form of ellipsis which occurs only in discourse in which (at least) two agents participate in a dialogue (see dialogue excerpt (55)). This type of dialogue based ellipsis is also called *fragments* (see Schlangen and Lascarides (2003)). The following example from the SMARTWEB system illustrates the usage of elliptical constructions:

- (55) **User:** “*What’s the weather forecast for today?*”
System: [Displays the weather forecast]
User: “*And for tomorrow?*”

This ellipsis can be termed best as *discourse ellipsis* since speaker A relates her second request to her previous one by skipping all constituents that have already been mentioned. Especially in information-seeking dialogues—like searching for a movie with weak preferences—a user might often try to change only parts of his previous utterance without repeating the whole utterance.

Additionally, elliptical constructions often occur in the context of question—answer-pairs. Again, aforementioned constituents in the question can be skipped as in the example from the SMARTWEB scenario:

- (56) **System:** “*Where do you want to go to?*”
User: “*Berlin.*”

Moreover, elliptical expressions sometimes do not simply replace a previously mentioned constituent. In fact, they first have to be interpreted with respect to their context before the actual ellipsis can be resolved. Consider for example the following fragment from the OMDIP system:

- (57) [A list of available songs is displayed on the screen]
User: “*Play the first song.*”
System: [Plays the first song]
User: “*And the second.*”

In this example, the elliptical expression “*And the second*” itself consists of an exophoric referring expression that needs to be resolved. Subsequently, the resolved reference can be

integrated into the previous request. The same phenomenon also occurs with referring expressions referencing referents of different types:

- (58) **User:** “How many students are enrolled at Saarland University?”
System: “About 15.000 students.”
User: “And in computer science?”

This type of ellipsis is called *expansion ellipsis* and has received considerable interest in the area of natural language dialogue systems already in early years (see, for example, Allen (1979); Jameson and Wahlster (1982); Weischedel and Sondheimer (1982)).

Schlangen and Lascarides (2003) present an empirical validated taxonomy of *sentential fragments* (i. e., discourse and question ellipsis). Schlangen and Lascarides classify fragments along two dimensions. In the first dimension they distinguish between fragments that can be resolved by identifying the missing content in the previous discourse (they call this type of resolution *resolution-via-identity*), and fragments that can only be resolved by inference from both the linguistic sources and the extra-linguistic context.

In the second dimension they categorize fragments by their discourse function with respect to the context. Schlangen and Lascarides classify the fragment in example 56 as a *Question-Answer Pair* (QAP). In total, their taxonomy comprises 25 different types of fragments. Schlangen and Lascarides (2003) tested the coverage of their taxonomy by analyzing 9124 items from two corpora. In total, these items comprised 931 fragments (= 10.2%), of which they were able to classify 865 (= 93%). But, as the authors note, the distribution of the different types of fragments is concentrated in a small subset of the available classes while the rest of the classes appear only in a few examples.

Schlangen (2003) provides an extended taxonomy of fragments. This taxonomy covers all types of speech acts that can be performed with non-sentential utterances (NSUs). The main difference is that it also comprises fragments that do or do not directly convey messages or contribute to the propositional development of the conversation. Thus, short conventionalized utterances like “Bye” or backchannel feedbacks are also treated as individual classes of fragments. Table 3.10 shows an excerpt from the frequent classes of this extended taxonomy (including the distribution values) as presented in (Schlangen, 2003)⁷.

When considering the frequency of the different classes of fragments, one sees that the majority of fragment instances is concentrated in a few classes. The rest is only represented by a small number of examples and thus does not play an important role in every day communication. The most frequent type is QAP (29,8%) followed by backchannel feedback (Backchn) (25,2%), by *Elap_{pq}* (12,1%) and Agree (12,0%), while the remaining classes show only low frequencies. However, it is important to keep in mind that Schlangen employs a very fine distinction by splitting usually jointly grouped fragments into different classes.

3.2.5 Common Ground

Common Ground is the basis for any spoken interaction between two or more participants (Stalnaker, 1978; Clark, 1996; Traum, 1994). The common ground of two people is the sum of their mutual beliefs about the conversation and the surrounding world. As Clark (1996) highlights, when strangers meet, they start building up presuppositions about the

⁷In Schlangen (2003) the distribution values are provided separately for the two corpora that were used in the study. However, since we are not interested in individual effects of the corpora, we have summed up these values.

Class	Definition	Example	Freq
<i>QAP</i>	β provides a direct answer to α	<i>A: Who came to the party? — B: Peter</i>	258
<i>QAP_q</i>	β is a y/n-question, and positive answers to it provide a direct answer to α , negative answers a partial answer.	<i>A: Who was this? Peter? B: Peter.</i>	8
<i>Elap_{pp}</i>	β elaborates on some aspect of the indicative α	<i>A: I talked to Peter. Peter Miller.</i>	93
<i>Elap_{qp}</i>	β elaborates on the intended meaning of α	<i>A: Who did you talk to? Yesterday.</i>	47
<i>Elap_{pq}</i>	Any answer to β elaborates on some aspect of the indicative α	<i>A: I talked to Peter. - B: When?</i>	105
<i>Elap_{qq}</i>	Any answer to β elaborates on the intended meaning of α	<i>A: Did you talk to Peter? B: Peter Miller?</i>	23
<i>Contr</i>	α and β have a contrasting theme	<i>A: Are they in the cupboard? B: No, in the fridge.</i>	6
<i>Cont</i>	β continues a topic of α	<i>A: I am free on Monday. And on Wednesday.</i>	54
<i>Q-Cont</i>	The question β continues a topic of the question α	<i>A: What's his name? B: ... A: His address?</i>	30
<i>Q-Alt</i>	Answers to β answer an alternative question combined out of α and the fragment-phrase β	<i>A: Can you come on Tuesday? Or Wednesday?</i>	0

Table 3.10: Speech-Act types of *message-type* non-sentential utterances (fragments); β is the fragment and α the utterance to which it is related (adapted from Schlangen (2003)).

background knowledge of the other and thereby try to mutually update the common ground. If, for example, two people previously unknown to each other discover during small talk that they like the same musician, the style of their interaction will immediately change. They will use a different language that is only comprehensible for insiders or experts. What happens in such a situation is that their common ground is extended with domain specific knowledge, and both know that they can safely incorporate this knowledge into their contributions.

Communal Common Ground

When we encounter people we categorize them by nationality, profession, religion and so on. We use these categories of cultural groups to infer what they know, believe, or might assume. But the information people have about a certain cultural group depends on whether they are members of that group, i. e., insiders, or not, i. e., outsiders (Clark, 1996). Thus, inside information is information that members of a group mutually assume is possessed by insiders. In contrast, outside information is information that outsiders assume is insider information in a community.

Members of cultural groups are people with a shared expertise that members of other groups do not necessarily have. What makes them have a feeling of belonging together is their shared system of beliefs, practices, nomenclature, conventions, values and skills. Table 3.13 provides some examples of such groups and their shared expertise. Cultural groups have another interesting feature as they are typically not isolated from each other but rather exhibit

Class	Definition	Example	Freq
<i>Expl</i>	β explains α	<i>A: Peter left early. Exams.</i>	4
<i>Expl_q</i>	All answers to β explain α	<i>A: Peter left early B: Exams?</i>	30
<i>Expl*_q</i>	All answers to β explain why α has been uttered	<i>A: Are you married? B: Why?</i>	3
<i>Res</i>	α explains β	<i>A: He had a stroke. And died.</i>	4
<i>Res_q</i>	Answers to β are explained by α	<i>A: He had a stroke. B: And died?</i>	0
<i>Plan-Elab</i>	β details a step in a plan to reach a goal behind α	<i>A: Let's meet on Monday. At two o'clock.</i>	63
<i>Q-Elab</i>	Answers to β detail a step in a plan to reach a goal behind α	<i>A: Let's meet on Monday B: At two o'clock?</i>	29
<i>Ack</i>	β entails that B has accepted or achieved A's goal behind uttering α	<i>A: Let's meet on Monday B: OK</i>	78
<i>Plan-Corr</i>	β indicates that B does not accept or is unable to help achieve A's goal behind α	<i>A: Let's meet on Monday. B: No.</i>	7
<i>Ack_q</i>	Positive answers to β entails <i>Ack</i> , negative <i>Plan-Corr</i>	<i>A: Let's meet on Monday. OK?</i>	22

Table 3.11: Continued: Speech-Act types of *message-type* non-sentential utterances (fragments); β is the fragment and α the utterance to which it is related (adapted from Schlangen (2003)).

hierarchical relations. Through these hierarchical relations between groups, the assumed expertise associated with one group is also associated with all groups further down in the hierarchy. A New York resident living in Manhattan, for example, inherits all the assumed expertise of groups like North Americans, Americans, Easterners, or New Yorkers.

Personal Common Ground

Personal experiences have a great impact on our common ground. These experiences can either be shared perceptions of events (e. g., looking at the same object) or joint actions like talking. Clark (1996) calls them *joint perceptual experiences* and *joint actions* respectively. In order to achieve joint perceptual experiences, the participants must manage to attend to the same thing. This is achieved through salient events that lead each of them to assume that the others are sharing their own attention. Clark (1996) describes three main ways that joint salient events get established (Clark, 1996, p.113):

- 1 *Gestural indications*—Pointing gestures usually cause the addressee to focus on the object pointed at, which in turn enables the speaker to infer that this object is now part of our common ground.
- 2 *Partner's activities*—If the speaker notices that a participant is somehow focusing on a particular object (e. g., the addressee looks at a painting), the speaker can easily add this object to the common ground by referencing it.
- 3 *Salient perceptual events*—Distinctive perceptual events, like loud screams or unusual striking physical events like a piano that fell on the sidewalk, draw the attention of

Class	Definition	Example	Freq
<i>Comnt</i>	β indicates a propositional attitude of B towards the content of α	A: <i>I talked to Peter.</i> B: <i>Awesome!</i>	47
<i>Comnt_q</i>	Answers to β indicate a propositional attitude of A towards the content of α	A: <i>I talked to Peter.</i> B: <i>Really?</i>	2
<i>Narr</i>	e_β occurs after e_α	A: <i>He went to Italy.</i> And (then) to Spain	
<i>Narr_q</i>	Answers to β entail <i>Narr</i>	A: <i>He went to Italy.</i> B: <i>And then?</i>	
<i>Agree</i>	β indicates that B believes or agrees with the content of α	A: <i>Peter is an idiot.</i> B: <i>Yes.</i>	104
<i>Backchn</i>	β indicates that B understood α , or simply is still attentive	A:... B: <i>Hmm</i> A:...	218
<i>Pardon</i>	β indicates that B did not understand α	A: <i>Did Peter call?</i> B: <i>Sorry?</i>	62

Table 3.12: Continued: Speech-Act types of *message-type* non-sentential utterances (fragments); β is the fragment and α the utterance to which it is related (adapted from Schlangen (2003)).

anyone who is able to perceive them. Thus, if my attention is caught by such an event I can also infer that the attention of the people around me is also caught which means that event is now part of the common ground.

However, perceptual events can cause discrepancies in the common ground, as those events do not enter the common ground raw, but interpreted. If the speaker and the addressee do not share the same vocabulary, the addressee might fail to interpret the event the right way.

Extending the Common Ground—The *Grounding* Process

Establishing and extending common ground or mutual beliefs in human-human communication is a complex process that depends a great deal on acknowledgment. If some concept has been introduced into the discourse by a speaker, the listeners are supposed to provide some feedback about whether they think they have understood the contribution correctly. However, there is no one-by-one correspondence between contributions and feedback. In fact, the absence of negative feedback is often considered positive feedback.

Clark and Schaefer introduce with the *Contribution* model of grounding (Clark and Schaefer, 1989) one of the first models for the achievement of mutual understanding. The key idea of this model is the augmentation of the common ground through *contributions* that are composed of two parts: (i) the content of the speaker specifies the content of the contribution that has to be recognized by the listeners and (ii) the speaker and the listeners try to reach the *grounding criterion* which is a state where the speaker and the listeners mutually believe that they have understood what the speaker meant. Clark and Schaefer describe the two phases of a contribution as follows ((Clark and Schaefer, 1989, p. 265); **A** is the speaker, **B**

Category	Examples of groups	Examples of expertise
Nationality	American, Dutch, German	cultural practices, civil institutions, government of a nation
Education	psychology students, high school graduates	technical knowledge, educational practices
Employment	Ford auto workers, Stanford faculty, Newsweek reporters	facts about employer, colleagues, company practices
Language	English speakers, Japanese speakers, German speakers	phonology, morphology, syntax, semantics, lexicon

Table 3.13: Examples of communal groups and their shared expertise (adapted from Clark (1996)).

the listener):

- **Presentation Phase:** **A** presents utterance **u** for **B** to consider. He does so on the assumption that if **B** gives evidence **e** or stronger, he can believe that **B** understands what **A** means by **u**.
- **Acceptance Phase:** **B** accepts utterance **u** by giving evidence **e'** that he believes he understands what **A** means by **u**. He does so on the assumption that, once **A** registers evidence **e'**, he will also believe that **B** understood him.

When both phases are completed, it is common ground between the speaker and the listener that the listeners have understood what the speaker meant. Clark and Schaefer distinguish five main methods of conveying evidence of acceptance (ordered from the weakest to the strongest) (Clark and Schaefer, 1989, p. 267):

- **Continued attention**—The listener shows continued attention and thus signals the speaker that she is satisfied with the presentations.
- **Relevant next contribution**—The listener takes up what the speaker started and begins with a relevant next contribution.
- **Acknowledgment**—The listener provides acknowledgment by nodding or uttering some *continuer* like “*uh-huh, yeah*”, or an assessment like “*that’s great*”.
- **Demonstration**—The listener demonstrates all or parts of what she understood, e. g., through *reformulating*, *paraphrasing*, or *completing* the speaker’s utterance.
- **Display**—The listener **displays** all or parts of the speaker’s last utterance verbatim.

Another aspect of the grounding process, though negative, is a *request for repair*. When listeners fail to understand the speaker’s contribution, they can request for repair as for instance in the following dialogue fragment:

- (59) **A:** “*What about tomorrow?*”
B: “*Huh?*”
A: “*What about tomorrow?*”

Label	Description
Initiate	Begin new discourse unit; content separate from previous uncompleted discourse units
Continue	Same agent adds related content to open discourse units
Acknowledge	Demonstrate or claim understanding of previous material by other agent
Repair	Correct misunderstandings of discourse unit content
Request Repair	Signal lack of understanding
Request Ack	Signal for other to acknowledge
Cancel	Stop work on discourse unit, leaving it ungrounded and ungroundable

Table 3.14: The seven *Grounding Acts* of Traum's *Grounding Acts Model* (Traum, 1994).

In order to avoid infinite recursion in accepting phases, Clark and Schaefer postulated the following *strength of evidence* principle: If evidence e_0 is needed for accepting utterance u , and e_1 for accepting e_0 , then e_1 will be weaker than e_0 .

Traum discusses the main deficiencies of Clark and Schaefer's contribution model and compares it to his own computational approach to grounding, the *grounding acts model* (Traum, 1999, 1994; Traum and Allen, 140). Traum criticizes, for instance, the fact that Clark and Schaefer do not provide a measure of how much acceptance is needed to complete grounding. Since every signal itself is considered a presentation that needs acceptance, the question is to what point utterances need to be accepted and what utterances can be left un-accepted. Moreover, Traum points out problems with the ordering of the graded evidence of understanding. Finally, Traum criticizes what he calls the *off-line nature of phases* (Traum, 1999, p. 125) that makes the implementation of the contribution model for a conversational agent difficult. Traum argues that Clark and Schaefer's model requires a look at large segments of the conversation, both before and afterwards, before one can decide how a particular utterance fits in.

Traum's model is an on-line reformulation of the contribution model (Traum, 1994). Unlike the two phases of the original model, this model builds on *grounding acts* as the basic building blocks. These are linked with particular utterance units and perform specific functions towards the achievement of common ground. In total, Traum distinguishes seven grounding acts (see table 3.14) that could add or change the content of *discourse units*⁸.

Additionally, the grounding acts are also distinguished as to whether they are performed by the initiator or responder of a unit. Based on the resulting thirteen⁹ grounding acts, Traum presents a transition network comprising seven states that enables a participant of a conversation to determine the grounding status without any need for clarifying subsequent utterances. However, as Traum points out himself, the grounding acts model still shows some deficiencies (Traum, 1999). Whether a discourse unit is grounded or not is still represented by a binary relation which is an oversimplification.

⁸Traum calls the units of grounded discourse *discourse units* rather than *contributions*.

⁹Obviously the *Initiative* grounding act can only be performed by the initiator and thus there are only thirteen instead of fourteen grounding acts.

Own turn	[backchannel]	I see, mhm, really, right, yes
	[re-opener]	really
	[follow-up]	right
	[<call-off>]	OK
Slot 1	<uptake>	oh, well, yes
	<answer>	no, right, sure, yes
	<reply>	I see, mhm, oh, right
	<closer>	right
Slot 2	<emphasizer>	that's right
Slot 3 & 4	<frame>	all right, anyway, now, OK, right
Slot 5	<appealer>	all right, OK, right

Table 3.15: Some examples of common lexical items that serve as interactional signals or discourse markers in various contexts and their functions (adapted from Stenström (1994, p.66)).

3.3 Interactional Aspects of Dialogue

So far, we have discussed only those aspects of discourse that are related to the propositional content of discourse. In this section, we will now discuss the interactional aspects of discourse with an emphasis on those phenomena that require access to contextual knowledge. First, we will briefly discuss findings concerning the interactional structure of discourse. Then, we will discuss three key aspects that affect the organization and control of evolving interactions, namely: turn-taking, addressee identification and backchannel feedback.

3.3.1 Introduction

Speakers frequently use certain items to start, continue and terminate conversations. These items only contribute to the interactional aspects of the conversation as they do not convey any meaning that contributes to the propositional content of an interaction. All items that themselves constitute turns or link turns together are referred to as *interactional signals* (Stenström, 1994, p. 16). In contrast, there is another category of items that function as turnholders within the turn, or as boundary markers. These are referred to as *discourse markers* (Stenström, 1994, p. 16).

The actual function of these items depends on their position within the turn and within the sequence of turns. This means a listener is not able to identify the speaker's intention through the form of the utterance. The actual meaning of an utterance is thus "a matter of WHEN and WHERE something is uttered, by WHOM and for WHAT PURPOSE" (Stenström, 1994, p. 43). Within an utterance, Stenström distinguishes five slots where interactional items and discourse markers can occur:



Table 3.15 presents some examples of common lexical items that can serve as interactional signals or discourse markers. The actual function of a particular lexical item depends on its

position within a turn and also on the position of the turn within the exchange.

Interactional Signals

Interactional signals are used to start, carry on and terminate the conversation. They typically appear at turn beginnings or endings and their function varies with their position. Consider, for example, the different functions of the interactional signal *RIGHT* in the following two dialogue fragments (all taken from (Stenström, 1994, pp. 61)):

(60) A: "... *it's under H for Harry.*"

B: "*RIGHT.*"

(61) A: "*I'll see how I go.*"

B: "*OK RIGHT.*"

In the first fragment, *RIGHT* serves as a response to the statement of the previous speaker while in the second fragment it rather serves as an emphasize.

Functions of Pauses

Contributions in conversations are usually not just a continuous flow of speech. In fact, they are often interrupted by short pauses or inaudible sounds (called *filled pauses*). The length of these pauses is relative and it varies according to the rhythm of each individual speaker (Stenström, 1994, p. 7). As Stenström continues, silent and filled pauses are used for partly the same and partly for different purposes. Silent pauses serve as breathing pauses (such pauses match semantic-syntactic boundaries). Both types of pauses (often combined) are used for hesitation and for strategic purposes (e. g, taking and yielding the turn), and to mark off units of discourse (e. g., topics and subtopics).

3.3.2 Turn-Taking

When people engage in conversation, they take turns speaking (Duncan, 1972). People are remarkably good at taking and releasing a turn and only few speaker-listener alternations result in a temporarily break down of the conversation. When, for example, two participants in a discussion try to take the speaking turn at the same moment, in most cases this results in overlapping, inaudible speech. However, not all overlapping speech causes a temporarily breakdown of the floor management as, for example, it is an expression of mutual understanding when a listener completes his sentence with the very same words. In general, the turn-taking system is a crucial part of human communication, as Sacks et al. (1974) conclude:

In sum, turn-taking seems a basic form of organization for conversation-'basics', in that it would be invariant to parties, such that whatever variations the parties brought to bear in the conversation would be accommodated without change in the system, and such that it could be selectively and locally affected by social aspects of context. (Sacks et al., 1974, p. 700)

But before we can describe the turn-taking system in more detail, we first need a better understanding of what a *turn* is exactly. According to Bußmann (2002), there are three ways to define the term *turn*¹⁰:

¹⁰English translation of the German original text.

- 1 Based on formal criteria: turn as being isolated by pauses or as syntactic unit after which a speaker-listener alternation is possible.
- 2 Based on functional criteria: a turn contains at least a single move in terms of Goffman's discourse analysis.
- 4 Based on conversational analysis: a turn is the product of a process whose length and structure is interactively determined.

At a first glance, definition 1 seems to be quite intuitive as it defines a turn based on its boundaries (pauses or syntactic units) after which a change of speaker can occur. This view is very popular but ignores the fact that in everyday conversation a speaker often receives feedback by the listener while continuing with the turn. To overcome this obstacle, a turn is often defined as a period of speech from one speaker without verbal contribution from the other. However, strictly speaking this means that every contribution, for instance, the expression of some feedback of a listener first of all has to occur at the precise moment the speaker has foreseen it, and second of all that it can only comprise nonverbal behavior, as verbal contributions would cause the alternation of the speaker and the listener. But as we will see in section 3.3.4, this kind of listener feedback occurs very frequently in everyday dialogue and is a crucial component of spoken interaction.

Models of Turn-Taking

One of the first comprehensive models of the turn-taking system was put forward by Duncan (1972). Duncan's work is centered around the idea that the turn-taking mechanism is managed by signals composed of "clear-cut behavioral cues," or turn-taking signals that are used and responded to according to a set of turn-taking rules. Based on a thorough transcription of speech and body motion behaviors of two dyadic interviews (recorded on videotape), Duncan identifies a set of six discrete behavioral cues that display a turn-yielding signal of a speaker:

- 1 intonation,
- 2 paralanguage,
- 3 body motion: termination of any hand gesticulation,
- 4 appearance of stereotyped expressions, like "but uh" or "you know",
- 5 paralanguage: pitch/loudness,
- 6 syntax: the completion of a grammatical clause.

Duncan (1972) also stresses that these signals do not appear on their own but rather appear in combination: "The emphasis on inclusiveness requires joint consideration of the linguistic, paralinguistic, and body motion components of face-to-face interaction, as opposed to focusing exclusively on any one or two of these modalities." This means that a computational model of the turn-taking system must consider all these modalities in order to be able to cover the complete variety of human turn-taking behavior. Moreover, Duncan identifies what he calls an *Attempt-Suppressing* signal which, when displayed by the speaker, suppresses any turn-taking attempts of the listener. Typically, this signal is present when the speaker engages one

or both hands in gesticulation. Duncan identifies several other turn-taking signals and rules which will be summarized in the following subsections.

Based on empirical observations, Sacks et al. (1974) put forward a turn-taking model called *simplest semantics for turn-taking*. By observing audio-recordings of human-human interactions, they have found that the turn-taking system can be described in terms of two components, the *turn-constructional* component and the *turn-allocation* component plus a set of rules. The following is a basic set of rules for smooth turn construction, allocation of the next turn, and for coordinating transfer as proposed in (Sacks et al., 1974, p. 704):

- 1 For any turn, at the initial transition-relevance place or at the initial turn-constructional unit:
 - a If the turn-so-far is constructed in a way so as to involve the use of a 'current speaker selects next' technique, then the party so selected has the right and is obliged to take next turn to speak; no others have such rights or obligations, and transfer occurs at that place.
 - b If the turn-so-far is constructed in a way as not to involve the use of a 'current speaker selects next' technique, then self-selection for next speakership may, but need not, be instituted; first starter acquires rights to a turn, and transfer occurs at that place.
 - c If the turn-so-far is constructed in a way so as not to involve the use of a 'current speaker selects next' technique, then the current speaker may, but need not, continue, unless another self-selects.
- 2 If, at the initial transition-relevance place of an initial turn-constructional unit, neither 1a nor 1b has operated and, following the provision of 1c current speaker has continued, then the rule-set a-c re-applies at the next transition-relevance place, and recursively at each next transition-relevance place, until transfer is effected.

Sacks et al. also identify a number of techniques by which the current speaker may select the next speaker, e. g., an addressed question like "*Peter, do you need some help?*" (see below). One of the major shortcomings of Sacks et al. is, as Ellis and Beattie (1986) discuss, the vagueness in the description of the feature which defines a transition-relevance place. When we consider everyday interactions, it becomes obvious that there are a lot of transition-relevance points but listeners do not attempt to take the turn at these points. It thereby follows that there must be some kind of signal—possibly intentional, possibly paralinguistic, or kinesic. What we need is some mechanism to signal and recognize the point in time when it is appropriate to change the speaking turn during a conversation.

Another shortcoming of Sacks et al.'s approach is that they only considered verbal cues and neglected the nonverbal aspect of face-to-face communication. Kendon (1990) has conducted the first study to analyze the role of nonverbal behavior in turn-taking in a systematic way. His focus is on the speaker's eye-gaze on the listener and he compares the responses of the listener when a contribution ended with or without speaker's eye-gaze. The main findings of this study with respect to turn-taking are that if an utterance ended with speaker gaze, 70.6% of the listeners responded without a pause, and 29.3% when an utterance ended without speaker gaze.

However, as Ellis and Beattie (1986) report, there are several problems with these results. First of all, the linguistic content of the conversation is completely ignored and there are only

two dyads that have been considered for the analysis of the turn-taking behavior. That is, the data provides only tentative results and limits the conclusion that can be drawn. Moreover, eye-gaze cannot serve as a major turn-taking signal since conversations that take place in *voice-only* conditions (e. g., telephone conversations) show no specific patterns of disfluencies or disruptions. Telephone conversations, in fact, appear to be smoother. But what has been noticed, is that speakers more often make use of filled pauses which help to hold the turn while planning the next part of an utterance. Ellis and Beattie (1986) report about four times more filled pauses in telephone conversations compared with face-to-face conversations. This means that speaker and listener seem to adjust their behavior to the specific conditions in telephone conversations to compensate for the loss of visual information.

Moreover, as Thórisson (2002) highlights, there is a type of prediction that also contributes to the turn-taking system besides the perception of information from signals. Thórisson compares calls this *anticipation* and he poses the hypothesis that the “[o]ppertunities for turn-transitions are identified using a mechanism of anticipatory perceptual processing” (Thórisson, 2002, p. 181). This means that participants make use of expectations in order to determine when it is appropriate to take the turn or produce backchannel feedback (see section 3.3.4).

Nevertheless, the studies of Duncan (1972) and Sacks et al. (1974) mark the beginning of a still ongoing research effort to explore the mechanisms of turn-taking. Many researchers have investigated the actual behaviors of which people make use to successfully participate in the turn-taking system. In the following subsection we will briefly consider some of these behaviors. The following considerations are based on a notion of the distinct perspectives the speaker and the listeners take in multiparty dialogue systems which can be found in (Pfleger and Alexandersson, 2004, pp. 343–344).

The Speaker’s Perspective

Listener provides Backchannel Feedback A listener provides backchannel feedback when or shortly after the speaker makes a pause. For the progression of the discourse, it is important to determine the type of the received backchannel feedback. If it is a clear positive feedback (e. g., repeated head nods, or a clear “*yeah*”), the corresponding discourse entities can be marked as being at least partially grounded and the speaker can go on with the contribution. If the feedback is classified as neutral (e. g., “*hmm*”), the speaker can go on but there is no clear effect on the status of the common ground. However, if the listener provides clear negative feedback (e. g., “*huh*”), the speaker must consider the immediately preceding contribution failed and should initiate a clarification sub-dialogue.

Listener Wants the Floor If the listener starts gesturing (e. g., raising a finger or a hand into the visual field of the speaker), or begins to frequently nod or shift their body posture, they make it clear that they want to take the floor. Suitable reactions are either to grant the floor by finishing speaking and looking at the listener, or to provide an *attempt-suppressing* signal, i. e., engaging one or both hands in gesticulation (Duncan, 1972). In the case of an attempt-suppressing signal, listeners will almost never take the turn.

Listener Refuses to Take the Turn If the speaker has just provided some turn-yielding signals but the listener does not want to take the turn, this is characterized by either the listener looking towards the speaker and remaining silent or by providing some backchannel

feedback characterized as *continuation signals* like “*hmm*”, “*yeah*”, sentence completions or brief questions for clarification (Duncan, 1972).

Listener Accepts the Floor If the speaker has just provided some turn-yielding signals and the listener is willing to take the speaking turn, the listener signals this by looking away and starting to speak (for short contributions they do not look away). If the speaker accepts this and remains silent, the transition takes place and speaker and listener change roles.

The Listener’s Perspective

Turn-Yielding Signals The speaker wants to transfer the turn. This is displayed by the speaker looking at the listener, terminating gesticulation and remaining silent. Some speakers also accompany those displays with a raise of eyebrows or by fixing their gaze on the addressee.

Turn-Holding Signals The listener just provided some turn requesting signal but the speaker wants to hold the turn. This is displayed by an attempt-suppressing signal (see above) and the listener will remain silent in most cases.

The Speaker Requested Backchannel Feedback The speaker has just paused after an installment and possibly glanced at the listener. In this case the listener is able to signal his or her understanding of the ongoing discourse and whether he or she wants to take the turn. What is actually signaled via the backchannel feedback depends on the current status of the grounding process (thus, on the interface of the discourse modeler).

Selecting the Next Speaker

Another important aspect of the turn-taking system is the selection of the next speaker, a task typically performed by the current speaker. The most obvious way to select the next speaker is perhaps a vocative like in example (62). However, as Lerner (2003, p. 189) stresses, there is a difference between summoning someone by name to select them as the next speaker, and merely addressing them. Consider, for example the following question of the moderator in the VIRTUALHUMAN system:

(62) **User:** “*Ms. Herzog, what do you think?*”

Besides using the vocative form to select the next speaker, there are several additional ways of selecting the next speaker. Especially for questions, it is sometimes sufficient to look at the intended next speaker. In some conventionalized situations, for example in classrooms, a simple pointing gesture can be used to select the next speaker. In the next section (3.3.3) we will consider this topic in more detail, albeit from a slightly different point of view, namely the identification of the intended addressee.

3.3.3 Identifying the Addressee(s)

The identification of the intended addressee(s) is an important task that participants of conversations with more than two speakers have to perform continuously (Clark and Carlson, 1982). The main question is how speakers signal who the actual addressee(s) of their contribution are so that all present listeners can easily determine on a moment-to-moment basis

who is supposed to react, i. e., take the floor, when the speaker has finished. To what extent a speaker needs to determine the addressee depends on the dialogue situation and the context within which an utterance takes place. Consider, for example, the question in (63) which leaves its addressee(s) open when viewed on its own.

(63) **John:** “*What’s his name?*”

However, when some contextual information is available—e. g., John and Peter are waiting at the elevator when a new colleague approaches; John glances at Peter while asking the question—then it is an easy task to infer that Peter must be the addressee of this question and that he is obliged to answer. But not only the situational context is important; discourse context also plays an important role. If we only slightly modify the situational context given above by adding another person waiting at the elevator and letting John glance at the approaching colleague, then it not clear anymore who is supposed to answer John’s question.

Explicit Addressing

Speakers usually tend to look at the intended addressees when they begin to speak, and to seek mutual gaze with the addressed participants in order to give them an opportunity to determine that they are being addressed. Sacks et al. (1974) state that a common method of *explicit addressing* is either to look at the addressee or to directly address him or her (i. e., through vocatives). Consider, for example, the following fragment from the VIRTUALHUMAN scenario:

(64) [Two human users are standing in front of the projection screen where the moderator and the two virtual experts are displayed]
Moderator: [looks at user 1] “*Ok, User 1, what do you think?*”

In this example, the moderator determines the intended addressee by looking at user 1. However, as Lerner (2003) stresses, gaze-directional addressing is vulnerable to the looking and glancing practices of participants. First of all, the gazed-at participant might not see that he is being looked at and second of all, the non-addressed participants might not see the gazing behavior of the speaker. Nevertheless, speakers are usually quite good at identifying these potentially ambiguous situations and make use of another explicit method of addressee identification, namely *address terms*. If the gazed-at participants show no attempt to accomplish mutual gaze, speakers, for example, can include a second person’s reference in order to make the participants to look at them.

As Lerner (2003) points out, when a speaker refers to a participant by using the name of that person or a reference like *he* or *she*, that person can be excluded from the list of potential addressees. But when a speaker adds a *vocative* or other address terms (e. g., categorical term of addressee *coach*) at the beginning (*pre-positioning*) or end (*post-positioning*) of the expression, this person remains the only possible addressee. Consider for example the following two discourse fragments:

(65) **User:** “*Ms. Herzog, what do you think?*”

(66) **User:** “*What do you think, Ms. Herzog?*”

Pre-positioned address terms are usually employed to verify that the addressed person is paying attention especially in acoustically problematic surroundings. Post-positioned address terms, in contrast, are typically used to further clarify who is the intended addressee if,

for example, a previous gaze-directional or other tacit forms of addressing did not have the intended effect.

Tacit Addressing

In this type of addressing, the recipients are supposed to draw upon diverse features of the specific circumstances, content and composition of an action to identify who is being addressed (Lerner, 1993, 2003). So, “when the requirements for responding to a sequence-initiating action limit eligible recipients to a single participant, then that participant has been tacitly addressed” (Lerner, 2003, p. 190). For example, a tacit method of addressing is, if the speaker poses a question that can only be answered by a single co-participant and the speaker knows this. Consider, for example, the following situation: Peter knows that Paul watched the baseball game last night while Mary did not. Then it is perfectly sufficient if Peter asks: “*Did the Red Sox win?*”. As Lerner (2003) notes, it is also possible to compose a turn that tacitly selects an addressee in a *negative* fashion by eliminating all but the addressed participants. Consider, for instance, the following situation: Two parents are standing in front of their children when the mother asks “*Will they ever behave?*”

Another often-applied form of addressing is that of sequential positioning of sequence initiating actions, as for example in follow-up questions where the speaker poses a sequence of questions to the same addressee (see for instance example (67)), or counter questions where the addressee takes up the speaking turn and poses a question to the previous speaker (see for instance example (68)). The following examples are taken from Sacks et al. (1974):

- (67) **Sy:** “*Did you see Death of a Salesman last night?*”
Jim: “*No.*”
Sy: “*Never seen it?*”
Jim: “*No.*”

- (68) **Peter:** “*Did you see the game last night?*”
Paul: “*No. Did you see it?*”
Peter: “*Yes.*”

3.3.4 Backchannel Feedback

Communication is not a uni-directional sequential process but rather a bi-directional highly interactional process (Clark, 1996). This means that in natural communication situations, not only a single speaker is conveying information but also the addressees and other listeners are constantly conveying information that influence the speakers’ behavior. This listener behavior is called *backchannel feedback* (Yngve, 1970) and can be expressed by means of both verbal and nonverbal behavior. Listeners can utilize a short pause by the speaker to provide backchannel feedback, thereby indicating that and to what degree they understood or agreed with what has been communicated so far (Yngve, 1970). It should be noted that speakers invite their listeners to do so by—sometimes unconsciously—placing pauses and possibly looking at the same time at the listener in order to obtain feedback.

The agreement or disagreement expressed by the listener has a direct effect upon the subsequent utterances of the speaker. An expression of puzzlement, for example, can cause the speakers to further clarify their intentions. In contrast, repeated agreement supports mutual understanding (*grounding*) of propositions or objects mentioned in the preceding

discourse (Clark and Brennan, 1991). Backchannel feedback can also be used to decline the opportunity to take the turn. Moreover, as pauses can be utilized by the listener to initiate a clarification sub-dialogue, speakers have to monitor the listener carefully to determine whether or not a repair is required. A listener can vary the degree of agreement either by varying the actual strength of the non-verbal expressions (e. g., “*hmm*” vs. “*yeah*” and slight head nods vs. emphatic head nods or facial expressions) or by the amount of time they take until they provide the backchannel feedback. Moreover, as overlapping speech causes no trouble (the listener can provide backchannel feedback while the speaker continues to speak), a thorough model of backchannel feedback must account for treating the communication channels as distinct entities that do not interfere. As a consequence, this implies that a simple “transmitter-receiver model will not be sufficient when transferring multimodal interaction to the computer domain” (Thórisson, 1996, p. 41).

Besides the turn-taking signals and rules mentioned earlier, Duncan (1972) has also analyzed backchannel behavior (what he calls “*listeners response*”). The according rule put forward by Duncan states that backchannel-behavior is used by a listener to avoid taking an offered turn and that it does not constitute a turn or even a claim for a turn. Backchannel feedback comprises a large and complex set of signals that are displayed in a variety of communication situations. Among the signals that Duncan (1972) identifies are:

- 1 para-verbal or stereotyped phrases, like “*hmm*” or “*yeah*”,
- 2 head nods,
- 3 sentence completions,
- 4 brief questions for clarification,
- 5 restatement in a few words of an immediately preceding thought expressed by the speaker.

Nakano et al. (2003) analyze eye gaze, head nods and attentional focus in the context of a direction-giving task. Their results show that the distribution of nonverbal behavior differs depending on the type of dialogue move being grounded. Moreover, they show that “the overall pattern reflected a *monitoring of lack of negative feedback*.” This supports the general observation that participants attempt to minimize the effort expended in grounding (see section 3.2.5) so that the absence of negative feedback equals positive evidence.

Discussion: The Impact of Backchannel-Feedback on the Grounding Process

As we have shown above, the extension of the common ground (*grounding*) requires an extended model of acknowledgment. Explicit acknowledgment is a rather uncommon way of providing acknowledgment unless the acoustic surroundings require it. However, constantly assuming mutual understanding without taking the possibility of miscommunication into account has been questioned by the supporters of profound models of grounding (e. g., Traum (1999)). But as Carlett et al. (2002) have put it:

The conversational strategy of assuming common ground only works because conversation is highly interactive, with participants able to alter the course of conversation almost immediately when they notice a problem. (Carlett et al., 2002, p. 4)

When we consider the frequency and location of backchannel feedback in conversations, it can be safely assumed that backchannel feedback is used by listeners to provide the required acceptance of presentations. As a consequence, Thórisson (1996) claims that “the success of the grounding process depends on the successful support of dialogue by the common organizational principles of turn-taking, back-channel feedback and other multimodal communicative mechanisms” (Thórisson, 1996, p. 38).

3.4 Conclusion and Discussion

In this chapter we have discussed the central context-dependent discourse phenomena that account for natural interactions. Even though the results presented here come predominantly from studies dealing with human-human interactions, we will assume that they can be transferred to the domain of human-computer interactions. This assumption is based on the notion of an intelligent agent as it can be found in textbooks on Artificial Intelligence (see, for example, Russell and Norvig (1995)). A *virtual agent*¹¹ is usually “viewed as **perceiving** its environment through **sensors** and **acting** upon that environment through **effectors**” (Russell and Norvig, 1995, p. 31) (original emphasis). Virtual agents substitute the human senses (e. g., eyes, ears) by means of sensors like cameras and microphones. In the same way, virtual agents substitute the human mouth, hands, legs and other body parts by means of speakers and various motors. But the important thing is that virtual agents are able to perceive their environment and act upon this perception. This distinguishes intelligent, autonomous agents from simple agents that act only on the basis of predetermined knowledge.

In this thesis, we will—on an abstract level—consider virtual and human agents as equal. This is, of course, an abstraction from the actual reality since virtual agents are not able to perceive their environment with the same level of detail as humans do. A virtual agent that participates in a dialogue with a human will not be able to perceive the clothing, the body language, or the social status of its human counterpart. Even though this information has impact on the development of human-human interactions, it is practically impossible for the virtual agent to recognize such aspects. The virtual agents have access to a limited symbolic representation of their environment as it can be recognized by the available recognition engines. The same symbolic representation is also used for inter-agent communication but here the recognition problems do not apply, which permits the representation of more complex information. In general, if the abstract symbolic representation comprises the right amount of information, virtual agents will be able to deal with the same discourse phenomena as humans do.

In this thesis, we are concerned with a multimodal fusion and discourse processing component that is part of a virtual agent that can participate in multimodal multiparty human-computer interactions. The ultimate goal is to develop full-fledged conversational dialogue systems that enable the users to engage in natural interactions with virtual agents.¹² Even though this goal might not be reached in the near future, it is still reasonable to incorporate

¹¹For the remainder of this thesis will use the term *virtual agent* to refer to artificial agents in contrast to a human agent.

¹²There are, however, authors that suggest to go the opposite way. Ahrenberg et al. (1995), for example, base their work on a sub-language approach that assumes that the language used when interacting with a computer will differ from the language between people. But we are confident that this form of reduced language is a result of the reduced capabilities of the currently available systems. The better the systems get, the more elaborately the users will interact with the systems.

as much natural discourse phenomena as possible.

In order to be able to deal with such diverse phenomena like referring expressions, turn-taking behavior, or backchannel feedback in human-computer interactions, a detailed context model is required that goes beyond the classic linguistic context provided by a discourse history. The requirements for the individual phenomena are in detail:

- **Anaphoric references:** A complete list of discourse entities that might serve as a referent for further referring expressions.
- **Indirect references:** A model of associated entities for each discourse entity (realized by means of a semantic network).
- **Place deictic/spatial references:** A representation of the current active frame of reference as well as a detailed model of the spatial relations that hold between the individual objects in the physical surroundings.
- **Time deictic/temporal references:** A representation of the current active temporal frame of reference and a model of special days/holidays.
- **Exophoric references:** A detailed semantic model of the physical surroundings.
- **Elliptical Expressions:** A detailed representation of the sequence of utterances including information about the speaker and dialogue act of each contribution.
- **Turn-taking behavior:** A detailed model of the present participants, their current role and the displayed interaction signals.
- **Addressee Identification:** A model of the conversational state encompassing information about the current and past conversational roles of the participants.
- **Backchannel Feedback:** A detailed model of the referential and grounding status of the mentioned discourse entities.

3.4.1 Developing the Argument for an Integrated Multimodal Fusion and Discourse Processing Component

In what will follow we will take up the discussion from the conclusion of the previous chapter where we started the development of our argument for an integrated multimodal fusion and discourse processing component. In this chapter we have discussed several discourse phenomena whose interpretation requires access to contextual information. A key phenomenon of multimodal dialogue systems is the use of place deictic references where a spoken referring expression can be combined with a pointing gesture. This type of referring expression is usually dealt with in a multimodal fusion component while all other referring expressions are processed by a discourse processing component. However, as discussed in section 3.2.3, it is also possible to use the same type of reference without an accompanying pointing gesture. Thus, the usage of these referring expressions is inherently ambiguous and an interpretation with respect to immediate and wider context of use is indispensable.

Moreover, when we consider a system that supports gaze behavior or other conversational gestures in addition to pointing gestures, the need for an integrated fusion and discourse processing component becomes even more apparent. Whether looking at another participant

is equivalent to a pointing gesture or functions as a turn-taking signal, depends on the context of use and cannot be resolved without considering the total communicative situation. To gain an unambiguous interpretation of most of the verbal and nonverbal behavior, each perceived monomodal event must be interpreted within its immediate context of use. As pointed out by Duncan (1972), participants indicate their respective intentions to move to the next unit of interaction (their degree of *transition readiness*) with signals. More strikingly, this means that participants not only have to take their own state into account, but also that of the others. Thereby, it follows that a comprehensive context model for multiparty discourse needs to maintain both a detailed representation of the participants of the conversation and of their current conversational role.

Chapter 4

Computational Approaches to Multimodal Fusion and Discourse Modeling

This chapter focuses on related work in the area of multimodal dialogue systems and provides an overview of computational approaches to multimodal fusion, context modeling and reference resolution for practical multimodal dialogue systems. Section 4.1 gives an introduction to the general characteristics of multimodal dialogue systems and provides an overview of common architectures of such systems. Section 4.2 presents related work in the area of context modeling. Section 4.3 gives an overview of common modalities employed in multimodal dialogue systems and presents key approaches to multimodal fusion. Section 4.4 deals with related work in the area of discourse modeling, while section 4.5 does the same for the tasks of turn-taking management and addressee identification.

4.1 Introduction

A number of outstanding multimodal prominent systems have been developed in recent years. Most of the systems are tailored to specific background applications which sometimes makes it hard to compare them. In the following, we will discuss the basic features of multimodal dialogue systems from an application-independent perspective. We will focus on two central aspects of those systems: (i) their architecture and in particular the information-flow and (ii) existing input modalities and multimodal interaction patterns.

In the following, we give a brief description of outstanding multimodal dialogue systems:¹

- The early and influential *Put-That-There* system of Bolt (1980). With this system the user can manipulate graphical representations of circles, squares and diamonds on a screen using (limited) speech and pointing gestures input.
- The XTRA (eXpert TRAnslator) system which is a multimodal interface to expert systems (Wahlster, 1991) that can be controlled by means of (typed) natural language and deixis input. The main task of the XTRA system is to support the user in filling out tax-forms.

¹Note that this is a non-exhaustive subjective selection.

- The Ymir system (Thórisson (1999)): Ymir is a framework of psychosocial dialogue that bridges multimodal perception, decision and action. It is used to realize autonomous characters like Gandalf, a character that is able to answer questions about the solar system (Thórisson, 1997).
- The QUICKSET system which realizes a multimodal interface for military planning activities (Cohen et al., 1997). The focus of QUICKSET is on map-based interactions where the user can control the system via voice and pen input or combinations of both.
- The MATCH (Multimodal Access To City Help) system which is a mobile city guide and navigation system (Johnston et al., 2001).
- The REA system (Cassell et al., 2001): REA is a virtual character that provides house descriptions in the real-estate domain. It has a 3D graphical body and communicates using verbal and nonverbal behavior with a single user.
- The MRE (Mission Rehearsal Exercise) system (Swartout et al., 2001; Traum and Rickel, 2002): Within the scenario of the MRE system, the user can interact with multiple virtual characters in an immersive environment.
- MAX (Multimodal Assembly eXpert) (Kopp et al., 2003): This system centers around a virtual character, Max, who is situated in a virtual environment. His main task is to engage in cooperative construction tasks, where Max multimodally demonstrates to the user the construction of complex objects and guides the user through the individual steps of the assembly procedures.
- The COMIC system (Boves et al., 2004; den Os and Boves, 2003), a dialogue system that supports the user in planning their bathroom and equipping it with tiles and furniture. A highlight of the COMIC system is that it is able to deal not only with speech and pen-based pointing gestures but also with handwriting and drawing as input modalities.
- The MIAMM system (Reithinger et al., 2005b): MIAMM is a multimodal dialogue system that supports the user to explore big databases (i. e., music databases). A highlight of the MIAMM system is that it supports haptic feedback.
- The SMARTKOM system (Wahlster, 2006) which is a multimodal dialogue system supporting a variety of applications such as EPG (electronic programming guide), controlling home devices, telephone or email. Key to SMARTKOM, is the situated delegation-oriented dialogue paradigm (SDDP), in which the user delegates a task to a virtual communication assistant (Wahlster et al., 2001).
- The EMBASSI framework (Herfet et al., 2001) which provides a goal-based multimodal interface for controlling a wide range of technical infrastructures. Similar to SMARTKOM, this goal-based interaction metaphor enables users to specify high-level goals for complex tasks and not to worry about the actual sequence of subtasks necessary to achieve the goal.
- The SAMMIE in-car dialogue system (Becker et al., 2006b): This system allows the user to control their in-car mp3-player by using speech and haptic interaction while driving.

4.1.1 Multimodal Dialogue Systems: Architectures and Information-flow

In this section we will discuss the key aspects of the information-flow in multimodal dialogue systems. Based on a discussion of standard architectures of multimodal dialogue systems, we will outline the requirements with respect to the information-flow when aiming for reactive dialogue systems. We will start with a consideration of multimodal reference architectures focusing on the individual components and a functional data-flow. Then we will discuss the extensions to the information-flow that are required for the realization of reactive behavior.

Multimodal Reference Architectures

In 2001, an international seminar on *Coordination and Fusion in Multimodal Interaction* took place at Schloss Dagstuhl in Germany. During this workshop, an abstract reference architecture for multimodal dialogue systems as depicted in figure 4.1 was developed (Bunt et al., 2005). This architecture modifies the original reference architecture of Maybury and Wahlster (1998) and envisions mechanisms for representation and inference on a broad range of models that can be accessed by all modules. It includes “models of the user (identity, capabilities, beliefs, and intentions) and other agents (e.g., system, software agents, intermediaries), a model of the discourse (to help track attention and information about interlocutor turns and also detect and correct errors), context (e.g., physical/spatial and temporal state), domain, task, applications and, of course, the media and modalities (their properties and any associated codes)” (Bunt et al., 2005, p. 330).

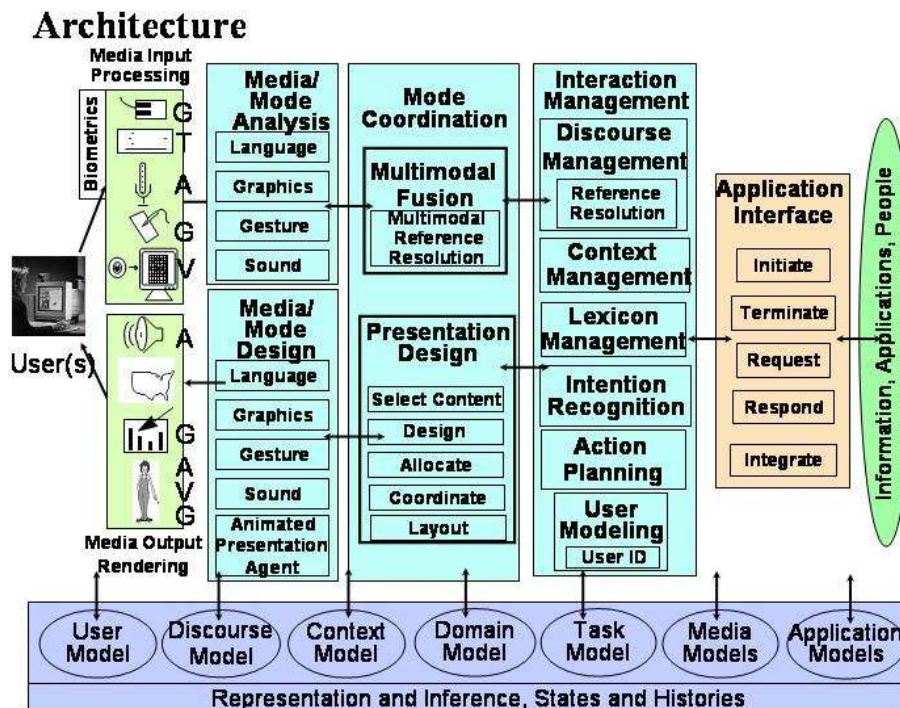


Figure 4.1: Extended Architecture for a multimodal dialogue system (from Bunt et al. (2005)).

Figure 4.2 shows a more abstract view of the individual components of a multimodal di-

dialogue system. Typically, an architecture of a multimodal dialogue system comprises three sequential processing phases: (i) an analysis phase (understanding), (ii) a dialogue management phase (action planning) and (iii) a generation phase (output planning and realization). Each phase can be realized by a set of standard components: (i) modality specific recognizers and analyzers, (ii) fusion and discourse processing components, a dialogue manager or action planner and (iii) modality fission, modality specific generators and realizers. Viewed on an abstract level, these phases can be realized by the components depicted in figure 4.2.

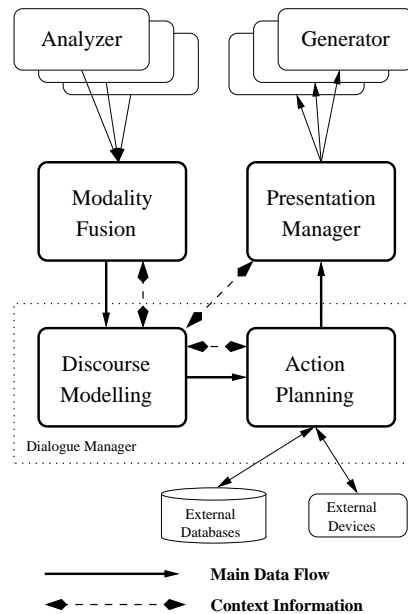


Figure 4.2: Abstract representation of the architecture and information-flow of a multimodal dialogue system.

The data flow is organized as follows: First, there are specialized recognizers for the individual modalities (e. g., speech and gesture but also recognizers for facial expressions) which produce the input for the individual analyzers. A natural language analyzer, for example, produces a set of intention hypotheses, where each hypothesis is a possible interpretation of the result of the speech recognizer. A gesture analyzer maps the gestures performed by the user to the objects presented on the screen and then publishes the set of referenced objects. The modality fusion module combines the resulting hypotheses of the analyzers and resolves the referring expressions that are accompanied by gestures pointing to an object on the screen. In the next step, the modality fusion module sends the enriched set of hypotheses to the discourse modeling component. This component maintains a representation of the discourse context and is able to resolve the remaining referring and elliptical expressions of the hypotheses.

The action planner (dialogue manager) selects the highest ranked intention hypothesis and executes an appropriate action, which is then visualized and uttered by specialized generation and visualization modules. The presentation planner (or turn planner) computes the high-level presentation and produces presentation orders which are realized by the text generator and the display device. Moreover, the action planner needs access to several external devices

representing the backend applications. Typically, backend applications are wrapped by some kind of a function modeler or semantic composer.

When we reconsider the discourse phenomena that are discussed in Chapter 3 it becomes clear that multimodal dialogue systems have to deal with a variety of challenges. To avoid overlapping speech or unexpected input, some systems simply keep the microphone closed unless the user is supposed to speak. However, this clearly restricts the interaction to a strictly user-driven or a strictly system-driven dialogue. Many Voice-XML-based systems², for example, keep the microphone open all the time. But then one has to deal with what is referred to as *barge-in* (the user speaks while the system is providing some output). Usually, VoiceXML-based systems handle barge-in by immediately stopping the ongoing output and continuing with processing the current user utterance. However, this strategy has the drawback that the system has no idea what the user is saying when it decides to stop its own contribution. This means that already short backchannel feedback might be interpreted as barge-in and cause the cancellation of the system's utterance. Moreover, there is another effect called *barge-before* which is "defined as the user reaction during the system processing but before the system output" (see Beringer et al. (2001)). In this case, immediately stopping an ongoing output activity would definitively be the wrong reaction.

The Information-flow in Reactive Dialogue Systems

Developing multimodal dialogue systems is a complex task, and therefore it makes sense to reduce the complexity through a pipeline-based processing architecture. However, when aiming for more natural and reactive systems that are able to deal with backchannel feedback or that can deal with multiparty situations, extensions to the information-flow are required. Enhanced and refined approaches have been put forward to overcome these obstacles (see for example Thórisson (1999), Allen et al. (2001)). In the following we will briefly discuss the approach of James Allen et al. and that of Thórisson.

Allen et al. (2001) have introduced the architecture depicted in figure 4.3 which tries to overcome the burdens of sequential processing architectures. Their core architecture consists of three main processing components: (i) the *Interaction Manager (IM)*, (ii) the *Behavioral Agent (BA)* and the *Generation Manager (GM)*. The Interaction Manager interprets user contributions and broadcasts the recognized speech acts and their interpretations as problem solving actions. It also updates the Discourse Context. The Behavioral Agent plans system behavior based on its goals and obligations, the user's actions and changes in the world state. Finally, the Generation Manager, plans and realizes the specific content of utterances and display updates. So far, this still resembles the standard architecture discussed at the beginning of this section. However, in contrast to the standard architecture, all three components operate asynchronously and do not depend on each other. So that, for instance, "the GM might be generating an acknowledgment while the BA is still deciding what to do" (Allen et al., 2001, p. 3).

Thórisson introduces a distributed modular architecture—called *Ymir*—that can be used to realize autonomous virtual characters capable of conducting full-duplex multimodal dialogues. This architecture is based on a dialogue model consisting of three layers or *channels* (see figure 4.4). Ymir consists of six main types of elements (Thórisson, 1996, pp. 91):

- 1 A set of semi-independent processing layers, τ .

²See <http://www.w3.org/TR/voicexml21/>.

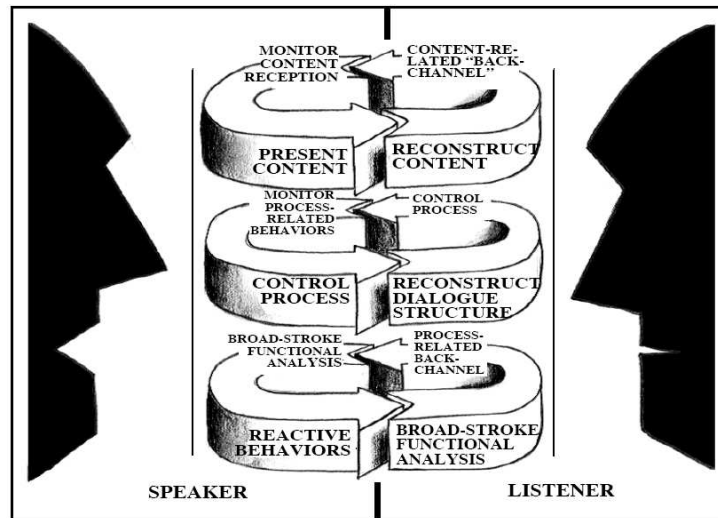


Figure 4.4: The three layers or *channels* of information transfer in multimodal communication (from Thórisson (1994)).

4.1.2 Designing Multimodal Interfaces—Human Factors

With the advent of multimodal dialogue systems, another important research aspect has come into focus: *Human factors in user interface design*. Pioneering work in that area has been conducted by Sharon Oviatt and her group (see Oviatt (1999b,a); Oviatt et al. (2003, 2004)). In (Oviatt, 1999b), Sharon Oviatt identifies 10 myths about multimodal interaction that were fashionable among system designers at that time and provides contrary empirical evidence for each of these myths.

Another important question is what the natural multimodal interaction patterns in human-computer interactions actually look like. It is likely that the artificial appearance of the systems and frequent misrecognitions cause users to employ a reduced and simplified language compared to the human-human communication situation. Boda (2004) reports on an analysis of a multimodal corpus comprising about 8000 utterances where the usage of deictic expressions in location reference patterns had been examined. Since the task of recordings for that corpus was a map-based one, 68% of these 8000 utterances contained deictic expressions. The results of this analysis showed that users tend to use “rather simple linguistic formulations and only the following ones occurred: *this, that, here, there, this one*” (Boda, 2004, p. 339). Boda could not find any occurrences of expressions like: *over there, over here, this university, that museum*.

Research concerning multimodal interaction also focuses on how people integrate their verbal and nonverbal contributions. In general, there are two types of integration patterns: *simultaneously* and *sequentially* integrated multimodal constructions. Oviatt et al. (1997) report on a study of multimodal interaction where people use speech and handwriting to control a simulated dynamic map system. They analyzed data collected from 72 map interaction tasks performed by eighteen native English speakers.

The following enumeration gives a brief summary of the main results of Oviatt et al.’s

study:

- **User Preference**—Subjects showed a strong preference for interacting multimodally during map tasks (100% of them used a combination of spoken and pen input during one task at least once). 19% of the constructions were expressed multimodally, 17.5% unimodally through writing, and 63.5% were expressed using only speech.
- **Task Action Analysis**—Spatial location commands that require a spatial location description are the user commands that were most likely expressed multimodally (86% of the multimodal constructions). Commands involving the selection of a specific object on the screen accounted for 11% of the multimodal constructions made by the users. However, it turned out that commands that involve the selection of an in-view object were more likely expressed unimodally because the object was already in focus from the previous dialogue context or the visual context. Other types of user commands were rarely expressed multimodally (only 3% of the multimodal constructions).
- **Linguistic Content**—98% of the multimodal construction conformed with the standard S-V-O order typical for English (compared to 98% of the unimodal spoken constructions). The prominent difference between spoken and multimodal constructions was the typical position of locative constituents. 96% of the spoken utterances showed locatives at sentence-final position, whereas in 95% of the multimodal construction locatives (using the pen) were first expressed and followed by spoken S-V-O constituents. Only 41% of the multimodal utterances contained a spoken deictic term.
- **Multimodal Integration Patterns**—86% of the multimodal constructions showed a draw and speak pattern whereas in 42% of the cases a simultaneous integration of drawing and writing, in 32% of the cases a sequential input, and in 12% of the cases a compound pattern took place. The remaining 14% of multimodal constructions showed a point and speak pattern. The synchronization patterns for the simultaneous integration of the draw and speak condition showed in 57% of the cases a precedence of writing and only in 14% of the cases a precedence of speech. Sequential integration patterns showed a temporal precedence of written input in 99% of the cases. The lag between the end of the writing and the start of speech averaged 1.4 seconds, with a maximum lag of 4 seconds. The integration of deictic terms showed in 43% of the cases a sequential integration pattern with an average lag of 1.1 seconds (a maximum lag of 3.0 seconds in 97% of the time).

4.1.3 Mutual Disambiguation and Error Handling in Multimodal Interfaces

As Oviatt highlights, “one major challenge for the design of multimodal systems involves learning how to combine different modes into a strategically integrated whole system” (Oviatt, 1999a, p. 576). The idea is that multimodal dialogue systems can benefit from the information on the different modes by combining them in a manner that permits mutual compensation. This means a multimodal system can avoid some of the recognition errors that would occur in unimodal system by evaluating whether the recognized monomodal events can be legally integrated or not (given some sort of semantic constraints that define well-formed commands or entities).

This approach is in particular helpful for the resolution of ambiguities that arise during the interpretation of the individual modes. As Wahlster (2002) pointed out, there is no universal algorithm for the resolution of such ambiguities. However, during the process of *fusing* the individual modes, some of the interpretation hypotheses can be ruled out when there is no matching counterpart in the complement mode. This process is called *mutual disambiguation* and is considered to be an important component of multimodal dialogue systems. This mutual disambiguation accounts for a significant increase in robustness and efficiency compared to classic speech-based dialogue systems.

4.2 Context Modeling

The notion of context plays a crucial role in a number of research areas such as computational linguistics (and here, especially pragmatics, e.g., Bunt (2000), Levinson (1983)), cognitive psychology and artificial intelligence (e.g., knowledge representation and reasoning (Bouquet et al., 2003)). The basic intuition underlying most work on cognitive processes is that reasoning in general is performed on only a subset of the global knowledge base. Or as Giunchiglia has put it:

[W]e never consider all we know but rather a very small subset of [the known facts]. This small subset is what determines the context of reasoning. We therefore take a *context c to be that subset of the complete state of an individual that is used for reasoning about a given goal.* (original emphasis) (Giunchiglia, 1993, p. 345)

Giunchiglia continues by drawing a clear line between context and situation where he defines a situation to be a representation “of the state of the world as it is, independently of how it is represented in the mind of the reasoner” (Giunchiglia, 1993, p. 345). Context, in contrast, is defined as a rather subjective view of a part of that world which is only available to the reasoning individual.

The general notion underlying the following considerations is that language understanding involves a context representation that changes successively as a result of the processing of the ongoing dialogue. This notion is closely related to the basic assumptions of speech act theory (see Austin (1962)) stating that linguistic behavior is action. Certainly, the *actions* of speech have no effects on the physical environment, it changes, however, the contextual representation of a receiver. This change of the receiver’s contextual representation includes, as Bunt points out “*that a communicative action operates on a given context to produce a new one*” (Bunt, 2000).

The major consequence of this view of context is that we have to interpret an utterance within its preceding context aiming at the enrichment of this utterance with contextual information. Finally, the resulting representation of this utterance can be added to the contextual representation, and can thus change the context.

4.2.1 Dimensions of Context

Before we discuss related work in the area of context modeling, we will briefly address the various dimensions of context in order to clarify what we understand by the term *context*. Ariel, for example, differentiates three types of contextual information that contribute to the resolution of referring expression (Ariel, 1990):

- **Encyclopedic or general knowledge.**
- **Physical context:** the physical surroundings of a dialogue.
- **Linguistic context:** previous utterances or text.

Bunt (1994) differentiates between *global* (static) aspects of context and *local* (dynamic) aspects of context. Following Bunt, global aspects have a certain value at the beginning of the discourse and tend to remain constant while the conversation is going on. The values of local aspects, however, are subject to change and have a “momentary significance in determining the continuation of the dialogue” (Bunt, 1994, p. 24). Based on these two aspects, Bunt defines five contextual categories: the *linguistic*, *semantic*, *physical*, *social* and *cognitive* contexts.

Even though these five categories sound quite reasonable, they are not sufficient to model the context of situated multiparty conversations. This is because Bunt’s work is focused on monomodal spoken dyadic dialogues and thus there was no need to incorporate the objects located in the physical context or more than two participants. In what follows, we will discuss an extension of Bunt’s context categories that overcome the limitations of the original categories.

Discourse Context

This category is equivalent to Bunt’s *linguistic* category. However, since discourse consists of more than just verbal contribution, the name *discourse* context seems more appropriate for this category. As we will discuss below, the main purpose of the discourse context is to support the processes of reference resolution and contextual enrichment.

Generally, a discourse context that is used for reference resolution must, at the very least, include the sequence of previous utterances and all accessible discourse referents (Poesio and Traum, 1995). Moreover, such context representation must also specify how an existing context is extended with a new utterance.

Situational Context

The situational context corresponds to Bunt’s global context and comprises situation-dependent default information that is static and mostly uncontroversial. This default information influences the interpretation of an utterance and is often left implicit. Situational context factors are *inter alia* the spatio-temporal properties of the situation or information both the speaker and hearer have about the underlying task. In contrast to these fixed factors, all other factors that can be affected by discourse belong to the remaining context categories.

Physical Context

As we will see below, Bunt has a different understanding of this category, as he views the physical context to be the “availability of communicative and perceptual channels; partners’ presence and attention” (Bunt, 2000, p. 100). The participants of situated dialogues, however, can refer both verbally and nonverbally to the surrounding objects. Thus, the physical context must also comprise a comprehensive model of the objects located in the immediate physical surroundings.

One of the first treatments of physical context in a dialogue system is the SHRDLU system of Winograd (1972). SHRDLU is a dialogue system that enables the user to control a small

world of objects (the BLOCKS world). But SHRDLU was also able to answer questions about the current configuration of the world and what was possible and what was not. Since SHRDLU is also able to resolve spatial relations, it has to maintain a representation of the spatial arrangements of the BLOCKS world.

Roy et al. use an implemented model *mental imagery* to represent the physical environment for robots (Roy et al., 2004). In their approach, a *mental model* is built up that represents a stabilized version of the physical reality. The mental model used for their robotic manipulator comprises representations of the robot's own body, a built-in model of the workspace table, a physical model of the body of the human communication partner, and objects situated on the work surface. Each object is described by its position, orientation, shape, color, mass and velocity (Roy et al., 2004, p. 1378).

Salmon-Alt also incorporates some aspects of the spatial organization of objects into her context model (Salmon-Alt, 2000). However, her model focuses on the creation of so-called *domains of reference* that support the resolution of referring expressions so that there is no central model of the physical environment.

Cognitive Context

In Bunt's original categories, the cognitive context is used to model the participants' states of processing and models of each other's states. To some extent this category seems redundant as most of this information is typically covered in the linguistic or discourse model. Bunt distinguishes the following global and local aspects of cognitive context (Bunt, 1994, p. 25):

- **Global aspects:**
 - (i) overall communicative goals of the participants,
 - (ii) the expertise of the individual participants concerning the domain of the discourse.
- **Local aspects:**
 - (i) current participants' beliefs, intentions,
 - (ii) plans for performing the underlying task,
 - (iii) the processing state of the participants (perception, interpretation, etc.),
 - (iv) the attentional states of the participants,
 - (v) currently active discourse topics.

Social Context

This category comprises information about the communicative rights, obligations and constraints of each participant. Again, Bunt distinguishes between the global and local aspects of social context (Bunt, 1994, p. 25):

- **Global aspects:**
 - (i) the type of institutional setting or communicative event in which the dialogue takes place (e. g., information-seeking, teacher-pupil or doctor-patient interaction)
 - (ii) the roles of the participants,

(iii) the relative social status of the participants.

- **Local aspects:**

- (i) the communicative rights and obligations that each participant has at a given point in the dialogue.

4.2.2 Computational Context Models

In the following, we will briefly discuss six computational approaches to context modeling.

A Multi-Purpose Model of Conversational Context

Poesio and Traum (1995) describe a unified general-purpose model of context. They discuss a model that supports the processes of reference resolution, speech act recognition and dialogue management.

The basis of their work is the Discourse Representation Theory (DRT) (Kamp and Reyle, 1993). Thus, the model is centered around a list of accessible discourse referents that serve as candidate antecedents for reference resolution. Poesio and Traum have extended the DRT model by including speech acts in the discourse model so that, in addition to the information about discourse referents, the model also comprises information about which speech acts have occurred and what the subordination relationships between them are. This means that their model, in contrast to standard DRT, is able to represent the fact that speaker *A* told *B* that *P*. As the authors claim, the proposal of adding speech acts to the discourse model “amounts to a shift from a model of sentence *meaning* to a model of sentence *use*.” (Poesio and Traum, 1995, p. 114).

This discourse model marks an important attempt to develop generic multi-purpose discourse models that can be employed for different tasks and systems than the one it has been developed for. However, its focus is on dyadic speech-based interactions and it remains unclear whether it can be applied to multimodal multiparty discourse without major modifications.

The COLLAGEN Dialogue Manager

COLLAGEN (COLLaborative AGENT) is an agent-based architecture for building dyadic dialogue systems (see Rich et al. (2001)). Key to COLLAGEN is a discourse state that is based on Grosz and Sidner’s theory of collaborative discourse (see Grosz and Sidner (1986, 1990)). COLLAGEN’s discourse state consists of a stack of goals called the *focus stack* and a *plan tree* for each goal on the stack. The top element on this stack is called the *current purpose* of the discourse. Plan trees in COLLAGEN are encodings of a partial shared plan between the user and the virtual agent.

The discourse state of COLLAGEN is updated after each action by the user or the system based on Lochbaum’s discourse interpretation algorithm (Lochbaum, 1998). Each discourse event is considered either (i) to start a new segment whose purpose contributes to the current purpose, (ii) to continue the current segment by contributing to the current purpose, or (iii) to complete the current purpose and popping it from the focus stack. Figure 4.5 shows an example configuration of COLLAGEN’s discourse state.

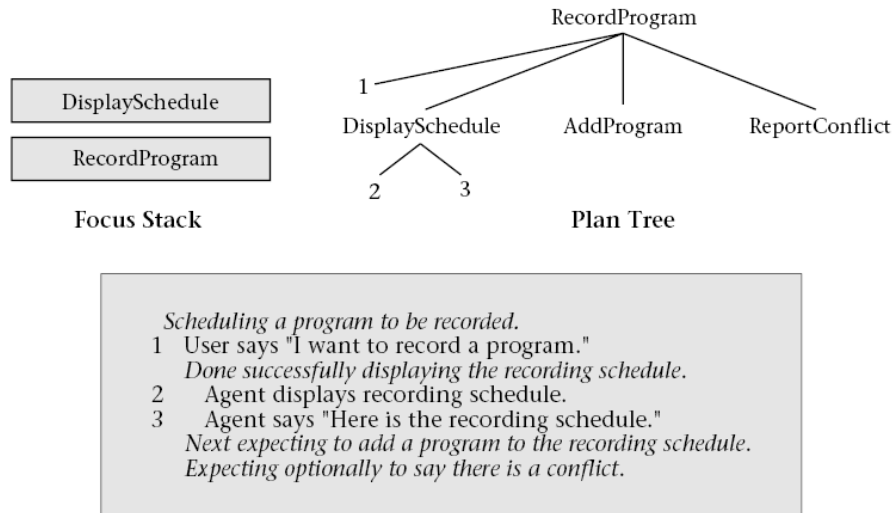


Figure 4.5: Example discourse state and segmented interaction for a COLLAGEN video cassette recorder agent (taken from Rich et al. (2001)).

A Multiparty Context Model

Traum and Rickel put forward a dialogue model that aims at modeling multiparty conversations in the Mission Rehearsal Systems (MRE). This model is organized on five layers (Traum and Rickel, 2002): (i) Contact, (ii) Attention, (iii) Conversation, (iv) Social commitments (obligations), (v) Negotiation. These layers are employed by the individual (artificial) conversational agents to build and maintain a representation of the ongoing conversation. Each layer encompasses both an *information state* representing the current status of that layer and a set of dialogue acts corresponding to the changes to the information state. Of particular interest for our purpose is the third layer which models the contextual aspects of a conversation. This layer comprises the following information:

- 1 Participants—active and overhearers,
- 2 The turn-holder,
- 3 The initiative-holder,
- 4 A dialogue history of utterances and actions that contributed to the conversation,
- 5 A grounding structure consisting of a set of *Common Ground Units* (CGUs).

However, Traum and Rickel (2002) admit that at the time of writing the model had only been partially implemented and based on the remarks specified in the paper, the actual processing strategies for common discourse phenomena remained unclear.

Dynamic Interpretation Theory

Bunt (1994, 2000) introduced the dynamic interpretation theory (DIT) which aims at developing effective dialogue systems. In DIT, context plays an important role as utterance

meanings are viewed in terms of context changes.

As already discussed in section 4.2.1, Bunt identifies five categories of contextual factors that form the *local context* of a dialogue (Bunt, 2000, p. 100):

- 1 **Linguistic context:** surrounding linguistic material, “raw” as well as analyzed. This is closely related to what is sometimes called “Dialogue History.”
- 2 **Semantic context:** state of the underlying task, facts in the task domain.
- 3 **Cognitive context:** participants’ states of processing and models of each other’s states.
- 4 **Physical and perceptual context:** availability of communicative and perceptual channels; partners’ presence and attention.
- 5 **Social context:** communicative rights, obligations and constraints of each participant.

Based on this rather broad notion of context, Bunt defines an iterative approach to context specification and dialogue analysis (Bunt, 2000, p. 101):

- 1 A set of utterance features is determined, to be interpreted pragmatically.
- 2 A set of communicative functions is chosen, often inspired by speech act theory.
- 3 A formal characterization of communicative functions in terms of context changes is developed, which often leads to reconsidering the system of communicative functions and corresponding utterance features.
- 4 To account for extensions and refinements in the functional treatment of utterance features, new or refined aspects of local context are introduced.

This means that Bunt’s approach consists of three sets of entities: (i) utterance features, (ii) communicative functions and (iii) local context aspects.

The TRIPS Discourse Context

The TRIPS system, as described by Allen et al. (2001) employs a rich discourse model that supports the analysis and generation of referring and elliptical expressions. It also comprises information about the current status of the turn, about who is holding the floor, and what discourse obligations are currently open. Allen et al. (2001) summarize the information the discourse model comprises as follows:

- A model of the current salient entities in the discourse.
- The structure and interpretation of the immediately preceding utterance.
- The current status of the turn—whether the floor is available or currently assigned to a participant.
- A discourse history consisting of interpreted utterances of the conversation so far.
- The current discourse obligations in order to be able to respond to the other participant’s last utterance.

Allen et al. claim that “this is a richer discourse model than found in most systems” (Allen et al., 2001, p. 3). However, it is still limited with respect to two important aspects: (i) it does not include the physical context of the discourse situation and (ii) it only supports dyadic dialogues (it is not possible to represent more than two participants—the user and the system).

4.2.3 Summary

Most context models discussed in this section focus on monomodal dyadic discourse except for the model of Traum and Rickel. A common limitation of all approaches is that they do not incorporate the physical context in which the dialogue takes place. As we will discuss later, there are discourse models that incorporate the organization of the physical context (i. e., the spatial representation of the objects in physical surroundings), however, only with respect to the resolution of exophoric referring expressions.

In 1991 Wahlster already emphasized the need for more elaborated context models:

It seems commonly agreed upon that a discourse model should contain a syntactic and semantic description of discourse segments, a record of the discourse entities mentioned, the attentional structure of the dialogue including a focus space stack, anaphoric links and descriptions of individual utterances on the speech act level. However, there seem to be many other ingredients needed for a good discourse representation that have not yet been determined in current discourse theory. (Wahlster, 1991, p. 48)

When we reconsider the phenomena of human-human interactions that have been discussed in Chapter 3, it becomes clear that a comprehensive context model would need to incorporate the complete conversational situation. These issues are partly addressed in Bunt (2000) and in Traum and Rickel (2002). However, the model of Bunt (2000) focuses only on monomodal (speech-based) dialogues and the model of Traum and Rickel (2002) seems only to incorporate the participants but not their current conversational roles.

4.3 Multimodal Fusion

As mentioned in section 1.1, a modality fusion component has to combine and integrate—if possible—all incoming unimodal events ($e_0 - e_n$) into a single representation of the intention that most likely has been expressed by the user. It is important that a fusion component makes sure that every unimodal event potentially contributing to the integrated meaning of a multimodal utterance ($m(e_0, \dots, e_n)$) will be considered. Thus, a fusion component needs to synchronize the recognition and analysis components so that all unimodal components of an utterance will be taken into account—a task often referred to as *multimodal synchronization*.

However, the task of a modality fusion component is not only to integrate the individual modalities but also, as Wahlster states, the reduction of “the overall uncertainty and the mutual disambiguation of the various analysis results” (Wahlster, 2003a, p. 9). Thus, a fusion component also needs to be able to deal with multiple interpretation hypotheses for the individual modes, and has to be able to determine the most promising integrated result. In this section we will first discuss common input modalities of multimodal dialogue systems, and then we will give an overview of the most prominent computational approaches to multimodal fusion.

4.3.1 Introduction

Multimodal dialogue systems usually provide access to a rich set of applications by giving the user the opportunity to employ various multimodal interaction patterns. This means that the user has the possibility not only to switch between the different modalities—depending on which modality is more suitable for the present situation—but also to use those modalities in an integrated and effective way.

Starting with Bolt's *put-that-there* system (Bolt, 1980), the history of multimodal dialogue systems shows a variety of different approaches. However, nearly all of the systems comprise a component that is capable of integrating the data streams of the different modalities into one combined representation (e. g., Cohen et al. (1997), Johnston et al. (2001), Wahlster (1991)). Basically, there are two major architectural approaches for the analysis part of multimodal dialogue systems: (i) early fusion—fusing modes already at signal or recognition level (see Bregler et al. (1993), Pavlovic (1998))—and (ii) late fusion—multimodal semantic processing (see Wahlster (1991), Johnston et al. (1997), Johnston et al. (2001)). Even though the point within the processing pipeline and the type of processed data vary, it is generally accepted that fusing the separate data streams will improve system performance by reducing uncertainty and resolving ambiguous input hypotheses. Another interesting aspect of multimodal interaction is that users can adapt their strategies of how to use the available modalities according to their current needs. According to Oviatt et al. (1997), users show a strong preference for multimodal commands and requests during map tasks. For example, if a user wants to set the starting point for the planning of a route, it is easier and more precise to use a pointing gesture towards the desired starting point on a map and accompany it with a spoken utterance containing a deictic reference (e. g., “*I want to start here*”) than giving “exact” coordinates by speech (e. g., “*I want to start about two and a half centimeters left from the church*”).

Wahlster (2003a) introduces the notion of symmetric multimodality for multimodal dialogue systems. This notion not only demands that all input modes (e. g., speech, gesture, facial expressions) are also available as output modes but also that the system must be able to understand both the user's input and its own multimodal output. This type of symmetric multimodality enables a multimodal dialogue system to deal not only with the simple integration of the modalities involved, but also with a variety of context dependent dialogue phenomena (e. g., anaphora, cross-modal (exophoric) references).

4.3.2 Input Modalities

An important aspect of multimodal dialogue systems is related to the input modalities that realize the interface between human users and machine. Besides speech, there is a great variety of input modalities that can be used for multimodal dialogue systems. In the following we will briefly discuss some of the common input modalities of multimodal dialogue systems.

Traditional Input Devices

The computer mouse and keyboard are still the most prominent human computer interfaces. However, there are several other devices that are commonly used as input devices. Some very popular ones are joysticks, handheld sticks that transmit their angle in two or three dimensions to the computer.

However, even though these devices are popular, they do not support natural and intuitive interaction between humans and computers. In fact, human users have to learn how to operate

these devices. Moreover, the interaction is not symmetric as the user and the computer have different means of communicating with each other. To this end, the multimodal research community tries to substitute these devices and focuses on new, more intuitive devices which will be discussed in the following subsections.

There are two devices that actually belong to the category of traditional devices but already show a more intuitive handling: (i) touch-screens and (ii) graphic tablets. Touch-screens serve as mouse replacements. The users can control the mouse pointer simply by touching the screen with their fingers. Touch-screens permit a more direct operation of the computer but they do not surpass the functionality of a mouse. Graphic tablets allow to hand-draw images directly into a computer, usually through an imaging program. The user can draw images on a flat surface using a pen-like device which is easier and more comfortable than using a mouse. However, the image does not appear on the surface of the device so that again there is no direct feedback of the user's actions.

In recent years, another class of input devices has emerged that is predominantly employed in cars. This type of input devices is often referred to as *controller knobs* (see figure 4.6 for examples). Usually, such a control knob can be pushed north, south, east, west, inwards, or rotated left and right. Additionally, such a control knob is surrounded by some buttons that permit to directly access a number of specific functions (e. g., entering the main menu). The BMW iDrive, for example, also provides haptic feedback (see page 93 in this section) so that the knob cannot be rotated further than there are options on the screen.



Figure 4.6: Three examples of in-car controller knobs for controlling secondary vehicle systems. From left to right: The BMW iDrive, the Audi MMI and the Mercedes Command.

Speech-based Interfaces

Speech is of course one of the most important modalities of multimodal interfaces since humans are very efficient in conveying information through speech, and consider speech to be their primary communication channel. In a multimodal dialogue system speech is typically recorded by a microphone and recognized by an automatic speech recognizer (ASR). Thus, speech recognition systems are able to capture a spoken utterance by a human user and to extract its propositional content. Speech recognition systems are classified along the following lines:

1 *Discrete vs. continuous recognition*

Discrete speech recognizers require the user to make a short pause between each word,

whereas continuous speech recognizers allow for more or less natural input. Continuous speech, however, is not the same as natural speech. People still have to adapt a very precise, clear manner of speaking and avoid running words together. Even though users typically prefer continuous recognition systems, discrete recognition systems are better at recognizing non-standard speech like disfluencies. Today, nearly every ASR supports continuous recognition; only a few systems are still available that support a discrete recognition, e. g., dictation systems for physicians or lawyers.

2 *Speaker dependent vs. speaker independent*

A speaker dependent ASR system is specifically trained to the voice of a particular user so that spoken utterances of other persons are usually not or only less accurately understood. During the training session, the ASR adapts to the specific characteristics of the user at hand. Speaker independent ASR systems use general models of human speech so that they are able to deal with a great variety of speakers and even with non-native speakers.

3 *Language model based vs. n-gram models*

Language models (or grammars) define the set of acceptable (well-formed) input sentences for a speech recognizer. As a consequence, a language model based speech recognizer is only able to recognize exactly the sentences available in the language model. Variations or unexpected input sentences lead to misinterpretations. As such, they are mapped to the sentence of the language model that is most similar. N-gram based speech recognizers, however, follow a different approach. In general, n-grams are subsequences of n items from a given sequence, in this case subsequences of words in a corpus. This means, an n-gram based speech recognizer can deal with a greater variety of input sentences. However, it might recognize sentences that are not syntactically or semantically well-formed.

4 *Lexicon size*

A lexicon typically defines the words an ASR is able to recognize. All other things being equal, a small vocabulary is easier to recognize than a large one. However, in combination with a rather strict language model, the lexicon size becomes less important.

But speech not only provides the propositional information of an utterance, there is also additional *nonverbal* information embedded in the speech signal like prosodic features that support the process of understanding. Various prosodic recognizers have been developed during the last decade that serve different purposes, so that, for example, the distinction between On-Talk and Off-Talk can be improved by using a prosody recognizer (Siepmann et al., 2001). Other approaches focus on classifying speakers with respect to their age, gender, or emotions. The *AGENDER* approach of Müller (2006) (see also section 2.3.5), for example, is able to determine the age and gender of a speaker with a high accuracy. For a general overview of current approaches to speech recognition see Jurafsky and Martin (2000).

Pen-based Interfaces

In general, there are two types of hardware devices for pen-based interfaces: (i) classic graphic tablets and (ii) touch-screens that can be operated by using a pen (e. g., Pocket PCs, tablet PCs or the Cintec touch-screen).

Depending on the capabilities of the recognition software, we can differentiate the following types of pen gestures:

- 1 *Pointing*: Selecting a single object by pointing at it with the pen (resembles the traditional mouse click).
- 2 *Encircling*: Selecting a set of displayed objects simply by encircling all relevant objects.
- 3 *Handwriting*: Handwriting recognition has been a large research area in the last decade and it is still far from being solved. While first attempts focused on the recognition of isolated and simplified characters, it is now the overall goal to be able to recognize human handwriting in real time. However, the improvements in the last few years have been very promising and there are already several commercial products on the market that show very good recognition quality.
- 4 *Symbolic gestures*: Pre-determined symbols that can be drawn by using a pen. These symbols have an associated command that is executed when such a symbol has been recognized.
- 5 *Drawings*: Recognition of free drawings. The user can draw objects that are recognized and interpreted by the gesture recognizer. As for the symbolic gestures, the user is restricted to pre-determined drawings.

Vision-based Interfaces

Vision-based gesture recognition is the most natural way of gesture tracking since it does not involve any instrumentation of the user. Especially since it is possible to perform image processing in real time, this recognition approach is receiving more and more attention.

The area of vision-based interfaces breaks down into two main fields: (i) video-based and (ii) infrared-based recognition. Both fields have very active and productive research communities so that only a limited overview of the main research directions can be provided here. Today, 2D-camera-based recognition is no longer a hardware problem, as virtually every camera can be used (even so-called web-cams can be used—however, the better the camera, the better the results). This means that the development of intelligent algorithms is the current research focus³. However, for 3D recognition the development of hardware is still an ongoing research topic. A common approach there, is to use two cameras that are mounted in a way that the pictures are recorded from slightly different perspectives, just as it is the case with the human eyes. Besides traditional video cameras, infrared-based cameras are also used for 3D recognition.

The camera based approaches are used to recognize the following human behaviors:

1 *Facial Expressions*

Recognition of facial expressions is very complex. However, today there are a number of approaches that are able to recognize facial expressions on a reliable basis (see Chibelushi and Bourel (2006) for an overview).

2 *Head Movements Recognition*

Head movements, i. e, nodding and shaking are relatively easy to detect using infrared cameras.

³See (Marcel, 2002) for a detailed review of current approaches.

3 *Hand Gesture Recognition*

Hand gesture recognition was originally conducted using data-gloves (see below). However, nowadays a number of vision-based approaches have been put forward (e.g., see Sanchez-Nielsen et al. (2004))

4 *Lip-Reading*

Lip-Reading is often employed in combination with speech recognition in order to improve the recognition accuracy. A small camera is used to recognize the lip movements (see for example Sicconi et al. (2005)).

5 *Eye Movements*

Eye movements can be recognized using so-called *eye-trackers*. There are two types of eye-trackers: (i) *bright pupil* and (ii) *dark pupil*. The difference between the two is based on the location of the illumination source with respect to the optics. Also, the setups vary greatly. The main difference is that some eye-trackers are head-mounted and some operate remotely as they are placed in front of the user.

The actual recognition task breaks down into two levels: (i) an analysis phase where the image is pre-processed and (ii) a classification phase where the actual recognition takes place. During the analysis phase it is important to segment the image into meaningful and stable objects. To simplify the segmentation and especially the discrimination of the objects from the background, some approaches use passive markers like colored points or active markers like luminous diodes. Another important aspect of the analysis phase is the posture analysis that identifies the configuration of the body and hands for each image. The actual recognition phase then operates on a set of features that can be extracted from the results of the analysis phase. The tracing of the movement, acceleration, speed and polar speed are some of the widely used features. Dynamic Time Warping, Neural Networks and Hidden Markov Models are techniques used for classification tasks.

Among others, important drawbacks of vision based recognition techniques are (i) the computational complexity of the image preprocessing and the recognition and (ii) the dependency on the light conditions.

Navigation in Virtual Environments

Data Gloves Special devices have been developed to track and recognize hand configuration in particular for navigation in virtual worlds. Typically, in such scenarios the user is equipped with a helmet or a small display and numerical gloves. These gloves allow the system to track hand and finger movements and to reconstruct them in virtual reality. Finger movements are recognized by means of mechanical or optical sensors while the hand position and orientation are determined via electromagnetic or acoustic sensors.

There are a variety of gloves which can be used for the capturing of hand motions. The most important differentiation factor between the individual products is the number of joints sensed. A five sensor glove measures the curling of each finger but cannot differentiate the degree of movement per finger joint. Basically, the five sensor gloves record the curling of each finger from open palm to closed fist. Sixteen sensor gloves record each finger joint and add *adduction* (which means recording how close or far apart each finger is). Twenty-two sensor gloves measure motions of the palm, including more complex motions of the palms.



Figure 4.7: Example of a data glove: The 5DT Data Glove 16.

Figure 4.7 shows an example of a data glove, the 5DT Data Glove 16 of Metamotion⁴.

Data gloves enable an accurate and detailed tracking of hand movements. However, there are two important drawbacks: (i) the data is noisy, plus the acquisition frequency is weak and (ii) the user is connected to the computer through cables which clearly reduces the freedom of movement (Marcel, 2002). With the advent of wireless connections the second argument has lost a lot of its impact. The data gloves of Metamotion, for example, can be equipped with a wireless system that interfaces the glove with the computer via a radio link.

Beckhaus et al. (2005) introduce two additional physical interfaces for navigation in virtual environments: (i) the dance pad and (ii) a chair based interface. Both interfaces are controlled by means of the body so that the hands are free for other tasks.

Haptic Interfaces

Haptic interfaces are interfaces that provide some kind of tactile feedback. The basic idea is to provide the sensation of mechanical devices like buttons or switches. so that users can sense that their current action has some kind of effect. Moreover, haptic devices can also be used to provide additional information, for example, a button that signals the amount of information that will be shown if the button is pushed.

Two examples of commercial haptic devices are TouchSense from Immersion and the Phantom device:

TouchSense from Immersion: The TouchSense technology is based on the concept of traditional touch screens that can be operated simply by using the fingers. However, instead of feeling just the hard touch-screen surface, graphical buttons can seem to depress and release very much like actual buttons and switches so that the users experience a responsive reaction by the touch screen which leads to a more intuitive, natural, multi-sensory experience⁵.

Phantom Device from SensAble Technologies: The SensAble Technologies PHANTOM product line (see figure 4.9) of haptic devices enables users to touch and manipulate virtual objects. Using these devices, the user cannot only explore three-dimensional

⁴See <http://www.metamotion.com>.

⁵See the manufacturer's web-site <http://www.immersion.com/industrial/touchscreen/> for more information.



Figure 4.8: Example application of Immersion's TouchSense screen.

virtual worlds but also virtually feel the consistency and surface structure of objects. It is even possible to simulate the penetration of objects, e.g., the cutting open of an organ. The Phantom devices have been used in several research projects as for example in the MIAMM project (Reithinger et al., 2005b). In this project a total number of five Phantom devices were gathered to simulate a PDA-like music player with five haptic buttons.

In general, haptic devices support a bi-directional way of interaction. On one hand, the user can extract information about the environment and on the other hand, the user can manipulate (modify) the environment by using the same device. This allows a more natural human-computer interaction. However, as Landragin et al. (2002) point out, the haptic gestures concern virtual objects and the application has to define a coherent paradigm of interaction. This has the important consequence, as Landragin et al. (2002) continue, that users have to learn which gestures can be performed in which context. The most important drawback of haptic devices is their price: the more precise, the more expensive the device. Another drawback is that the type of interaction is not always that natural and intuitive for naïve users.



Figure 4.9: The Phantom Omni haptic device.

In recent years, however, a new type of haptic interface has emerged in the automotive sector like, for example, the iDrive control knob that is introduced on page 88 in this section. More recently, as part of the SMARTWEB project, a prototype of a haptic interface for a

motorbike has been developed. This interface is integrated into the handlebars of a motorbike (see figure 4.10). It is designed in a way that the user can easily operate it even with gloves on.

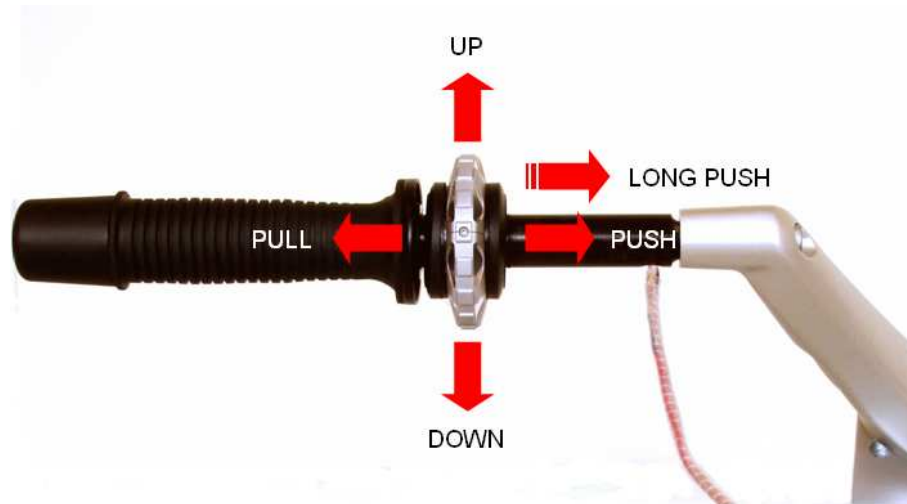


Figure 4.10: Prototype of a haptic interface that is integrated into the handlebars of a motorbike. The user can rotate the wheel, push and pull it.

Tangible Interfaces

Tangible interfaces are instrumented environments that are sensitive and responsive to the actions of users. In 1997, Ishii and Ullmer (1997) introduced their vision of human computer interaction called *Tangible Bits* which allows users to *grasp* and *manipulate* the objects in the virtual environment by coupling them with everyday physical objects and architectural surfaces. Thus, Tangible Bits is an attempt to bridge the gap between the physical and virtual environment. Ishii and Ullmer (1997) described three key concepts of their framework:

- 1 *Interactive surfaces*: Transformation of available surfaces in the environment (e.g., walls, desktops, windows) into active interfaces.
- 2 *Coupling of Bits and Atoms*: connecting graspable objects in the physical environment (e.g., books, cards, models) with their virtual counterpart.
- 3 *Ambient Media*: Integration of the periphery human perception, e.g., use of sound, light, airflow as an interface to the virtual world.

McGee et al. (2002) present an evaluation of a tangible interface called RASA (McGee and Cohen, 2001). Rasa is a system to support situation assessment in military command posts. It tracks the use of existing physical objects like Post-its and maps and recognizes multimodal actions by the users. The paper artifacts are overlaid on digitizers, so that any handwriting or drawing also appears in the digital system.



Figure 4.11: The tangible user interface COHIBIT. The virtual characters help the user to assemble a car on the workbench in the middle of the picture.

A more recent example of a tangible user interface is the COHIBIT⁶ car constructor (see figure 4.11), an interactive museum exhibit with virtual characters (see Ndiaye et al. (2005)). Guided by two virtual characters, the user constructs a car out of a set of car pieces. A central feature of the system is that the user can interact with the characters only by manipulating the tangible bits. Every time a physical building block of the car is moved, the characters will comment on that and encourage the user to continue. On the technical side, the tracking of the tangible objects is realized by means of RFID-tags that are attached to the 10 available car pieces. The table (workbench) where the user assembles the car comprises five areas where the pieces can be placed. The system is not only able to do simple detection work, but it can also identify which of the ten pieces is placed in each area of the workbench.

Another example of a tangible user interface is the *Mobile ShopAssist* (MSA) which provides shoppers with the possibility to converse directly with the products (Wasinger et al., 2005; Wasinger and Wahlster, 2006; Wasinger, 2006). As exemplified in figure 4.12, the user can query information about products—in this case digital cameras—via speech and gestures using a PDA. However, since the individual products are equipped with RFID-tags, users can also integrate physical actions (e. g., taking a product out off the shelf, or putting it back) into the interaction with the system. A more detailed discussion of this system is given in section 4.3.3.

In general, tangible interfaces provide an interesting way to incorporate physical actions into a multimodal dialogue system. In particular, these types of instrumented environments permit a more reliable recognition of physical actions compared to vision-based recognition techniques.

⁶COHIBIT stands for COntersational HElpers in an Immersive exhiBIT with a Tangible interface.



Figure 4.12: Sample interaction with the *Mobile ShopAssist* (MSA).

4.3.3 Computational Approaches to Multimodal Fusion

In the past decade, a number of approaches to multimodal fusion have been put forward. In what follows, we will discuss a selection of those approaches that influence our work.

Unification-based Multimodal Integration

The idea of unification-based multimodal integration goes back to the work of Wahlster in the XTRA system (Wahlster, 1991). XTRA is an early multimodal dialogue system that assists the user in filling out complex forms like income-tax forms. The key to this system was that it allowed the user to employ a combination of typed natural language and pointing gestures. XTRA uses a unification-based parser that is able to parse multimodal input. Pointing gestures are integrated into the input stream as terminal symbols that are then mapped to the preterminal category “deictic.”

Johnston et al. also make use of a unification-based approach for their QuickSet system. As Johnston et al. point out, the unification operation is best suited for the task of multimodal integration as it determines “the consistency of two partial information and integrates them into a single result” (Johnston et al., 1997, p. 284). Johnston et al. use the unification operation over typed feature structures to mediate and constrain the integration of speech and gestures. As the authors point out, this allows for the processing not only of deixis, but also of a wide spectrum of gestural input (e. g., pen drawings). In that, their approach resembles the one of Wahlster.

Holzapfel et al. (2004) present an architecture for the fusion of multimodal input for natural interaction with a humanoid robot. Their approach is also a rule-based one and bears some similarities to the multi-chart parser of Johnston et al. (2001). On the input-side, the

recognized monomodal input events are added to modality specific channels where they are transformed into a typed feature structure based semantic representation. Afterwards, the input events are added to a so-called input set which is used to synchronize the different recognition threads. To merge the different input streams, a constraint-based parsing approach is used.

The actual parsing is performed on a dynamic pool of tokens where elements can be added and removed. The parsing algorithm uses constraint definitions of so-called multimodal fusion rules to determine which rules can be merged. The parser supports a set of basic constraint types like content, time and modality constraints plus so-called *script* constraints which can be defined by the rule-writer. After the merging phase, special *construction* rules are applied in order to transform the tokens in the processing pool into the actual fusion results.

Holzapfel et al. (2004) thus introduce a flexible and generic approach to multimodal fusion that supports the separation of application specific and independent processing rules. However, a clear limitation of their approach is that they treat speech as the main or primary modality. This means that gestures are only used to resolve ambiguities, which clearly reduces the applicability of their approach in domains that require more than a simple point-and-speak interaction metaphor.

Multimodal Integration—A Statistical View

Wu et al. (1999) present a statistical approach to integrating the posterior probabilities of parallel input events in a multimodal dialogue system. Based on the observation that most previous approaches assumed that the individual modes in multimodal utterances function independently of each other, they argue that robust dialogue systems need to incorporate the posterior probabilities. Wu et al. (1999) give four factors that determine multimodal recognition performance:

- 1 recognition accuracy of the individual modes,
- 2 structure of the *associative map* (a projection between the indices of the multimodal classes and the indices of the individual modal output),
- 3 manner of combining posterior probabilities,
- 4 prior distribution of multimodal commands.

In general, Wu et al. state that “a multimodal system performs at its lower bound if individual modes are assumed to be independent, and a simple joint probability estimate is calculated during integration.” (Wu et al., 1999, p. 337).

Wu et al. (1999) integrated their approach into the Quickset system (Cohen et al., 1997) at that time. Quickset consists of two parallel recognizers for speech and gesture which are fused on a semantic level. The individual monomodal recognition results are integrated in three sequential steps:

- 1 *Temporally*

Quickset integrates speech and gesture input that is overlapped, or that appears within a specific time window when the signals arrive sequentially. Based on empirical research (Oviatt et al., 1997), Quickset employs a four second interval during which a gesture that follows speech will be integrated. Wu et al. (1999) note that the precise threshold

could also be learned by the system by using training data or could be pre-set for a specific domain by the developer.

2 *Statistically*

The original version of Quickset relied on the independence assumption; it used the cross product of the probabilities of the individual modes to compute the probability of each item in the final multimodal N-best list. The extended version of Quickset integrates the posterior probabilities and generates a N-best list for each final interpretation that includes posterior probabilities.

3 *Semantically*

Only those gestural and spoken elements in the N-best list that can be legally combined are integrated, which means that the resulting multimodal command must be executable by the system.

This extended version of the Quickset fusion approach introduces a new, statistical view that overcomes the independence assumption of the individual monomodal signals. However, this approach is limited in the same way as the original Quickset approach, as it also centers around only two modalities, namely speech and pen-based gestures. Another drawback is the limiting restriction that a spoken command is usually followed by a gesture. This restriction might be valid for the particular domain of the Quickset system (a generic system), however, has to be able to deal with more varying integration patterns (see Oviatt (1999b)).

Temporal Symbolic Fusion

Sowa et al. (1999) put forward an approach to multimodal fusion that is based on a common representation scheme for the different types of data and a rule-based integration mechanism. They employ a symbolic data representation that is organized as a concept hierarchy.

Sowa et al. (1999) employ a production rule system based on the CLIPS system (*C Language Integrated Production System*), which is a production rule system that directly supports the representation of complex knowledge. Sowa et al. (1999) highlight that a drawback of production rule systems in general is the complexity of the execution stage. The more symbols are available during rule matching, the higher the complexity of the execution cycle. To cope with this problem, Sowa et al. (1999) added a time window approach in order to keep the symbol memory small and efficient. The importance of a symbol decreases with time, and the system can remove the symbol if it does not have any relevance anymore. The time span of memorizing symbols depends on their degree of semantic content. Thus, more complex objects will last over a longer time span.

Multimodal Fusion in the SmartKom System

The SmartKom system (Wahlster, 2003a,b, 2006) enables the user to use speech and gestures in an integrated and efficient manner and provides wide coverage of different multimodal utterances expressing the same intention. This freedom of interaction leads to a broad set of challenging interaction patterns the fusion component has to deal with. SmartKom incorporates three modalities: Speech, gesture and facial expressions. However, the task of the fusion component is to combine only two of them, namely speech and gesture. Besides this, the fusion component also has the challenge to disambiguate the multiple hypotheses produced by

the different recognizers. Therefore, a score is computed for each integrated result based on a set of different parameters like temporal alignment, scoring of previous components and the number of replaced referring expressions. This score contributes to the final score for a hypothesis. The fusion component of SmartKom is discussed in detail in Engel and Pfleger (2006); in the following we will give only a brief sketch of the basic processing strategies.

The integration of pointing gestures selecting a single object on the screen is organized as follows: First, the natural language understanding component (Engel, 2006) assigns a special marker to each referring expression (called *refProp*), thereby indicating that such an object is not yet fully specified and needs to be enriched with information of an appropriate referent. Then, the fusion component tries to resolve those *refProps* by generating all possible combinations of *refProps* and recognized gestures that match the type of the expected referent. Consider for instance the following user utterance⁷:

(69) User: *I want to go from here [↗] to there [↗].*

In such a case, the output of the natural language understanding would comprise two embedded *refProps* of type LOCATION. The gesture recognition component would send a set of recognition hypotheses to the fusion component for the two pointing gestures containing the probable selected objects. Fusion, in turn integrates all hypotheses if possible (i. e. when the types of the objects match) and sends a set of integrated and scored multimodal intention hypotheses to the next component in the processing chain.

An interesting aspect of SMARTKOM's fusion component is the processing of encircling gestures. Encircling gestures are a specialization of pointing gestures. The user selects a set of objects displayed on the screen with one continuous gesture. In contrast to pointing gestures, encircling gestures require a somehow more elaborate processing in order to be able to correctly handle interdependencies with the previous dialogue context. Additionally, the integration of multiple objects requires world knowledge about the involved objects and processes to be able to integrate the encircled objects in a sensible way. Consider, for example, a situation where the system displays a seat reservation plan and asks the user to select the desired seats:

(70) User: *I want to reserve these three seats [encircling gesture].*

Here, the fusion component is typically faced with two problem sources: (i) the number of objects selected by the encircling gesture and referenced by speech does not necessarily correlate and (ii) for crowded displays of objects, it is likely that the organization of the objects (e. g., all objects in a row) produced by the gesture recognizer does not reflect the intention of the user. While the first challenge can be met by mutual disambiguation of the individual modalities, the second challenge requires more sophisticated consideration of the overall dialogue situation. To avoid presenting the user with a somewhat odd seat selection, SmartKom's fusion component applies a set of plausibility tests. These tests depend on the type of the selected objects and comprise, for example, a constraint stating that as many seats as possible should belong to the same tier.

Multimodal Fusion in Instrumented Environments

Wasinger et al. (2005); Wasinger (2006) present an approach to multimodal fusion of speech and gesture input in an instrumented environment. In their scenario, the user uses a mobile

⁷The arrow ↗ indicates the pointing gesture by the user when selecting an object on the screen.

shopping assistant (MSA) that integrates interactions on the mobile device and interactions with real world shopping products that are tracked via RFID-technology. Key to their work, is the differentiation between **intra-gestures**—on-device interactions, i. e., the user selects an object on the screen of the mobile device—and **extra-gestures**—off-device interactions, i. e., the user interacts with objects in the physical world.

Within this approach, Wasinger et al. define a modality-free language consisting of feature-object pairs that have a variety of attributes associated with them. Examples for features would be *optical zoom* or *mega pixels* in reference to digital cameras, while examples of objects would be *PowerShot S60* or *CoolPix 4300*. User input is only considered valid if it comprises at least one feature and one object.

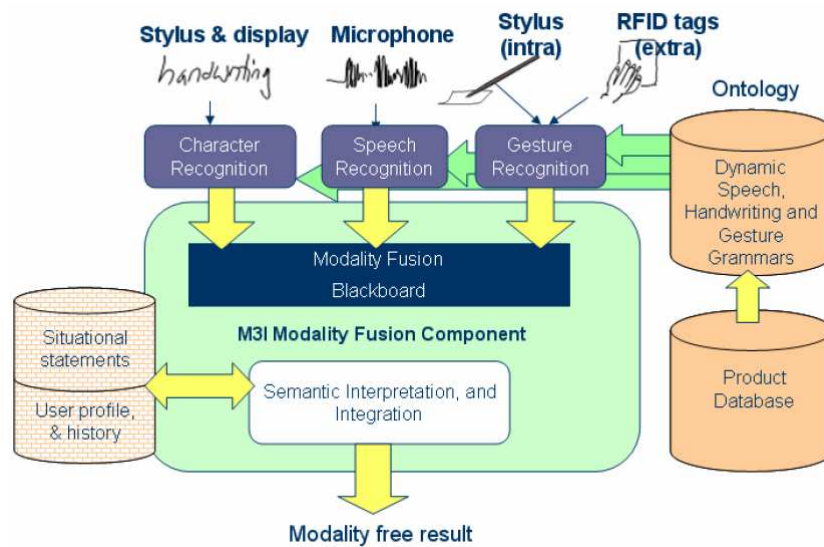


Figure 4.13: The architecture of the fusion component for the Mobile Shopping Assistant (taken from Wasinger et al. (2005)).

The key to their approach is that all the processing takes place on a mobile device (i. e., a PocketPC device), this also includes the recognition of speech and pen-input. All recognized input data is added to a central blackboard called *Modality Fusion Blackboard* (see figure 4.13), and the information is organized by means of *data notes*. A data note comprises information about the type of discourse segment (i. e., object or feature), the type of the input modality (i. e., speech or handwriting), the type of action where appropriate (e. g., point or pick-up), a confidence value, begin and end times of the segment, and several other pieces of information.

The actual modality fusion process is influenced by two parameters, (i) confidence scoring and (ii) timestamps which we will both discuss briefly.

Confidence Scoring Each modality specific recognizer produces n-best lists comprising the top three interpretation hypotheses and scores them with confidence values between 0.0 and 1.0. An important aspect of the work of Wasinger et al. (2005); Wasinger (2006) is that they developed a schema to compare the confidence of the individual recognizers. The key

idea of this schema is to re-weight the confidence values of the individual recognizers so that they are comparable with each other.

For re-weighting, Wasinger employs linear trend-lines of the form $y = mx + b$ where m and b are derived from a dataset that was acquired during a field study. Wasinger argues that this type of re-weighting confidence values is superior to simply taking the percentage of correct occurrences per confidence value “because the re-weighting approach accepts that neighboring confidence values are related to one another rather than being entirely distinct from one another” (Wasinger, 2006, p. 166).

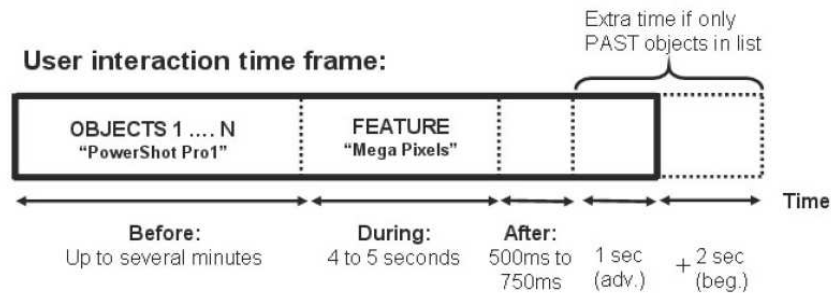


Figure 4.14: Temporal synchronization patterns of the Mobile Shopping Assistant (taken from Wasinger et al. (2005)).

Time-Frames and Input Synchronization As discussed earlier, users tend not to precisely coordinate their actions in different modes. Typically, a gesture occurs either before, during or after the accompanying spoken utterance. Concerning the temporal synchronization of input modes, Wasinger et al. developed an adaptive approach that permits incorporation of the user’s familiarity with the system (see figure 4.14). At a first glance, the individual time-outs look rather long compared to the 250 msec. within which a system response is considered to be reactive (see, for example, Raskin (2000)). However, the specific type of interaction where the users hold the PDA, its pen and possibly a product at the same time in their hands makes it necessary—in particular at the beginning of the learning phase—to give the users some extra time.

4.3.4 End-Of-Turn Detection

An important aspect of multimodal interaction is the precise determination of the end of a turn. However, the flexibility and the multiple ways a user can combine and integrate the different modalities pose serious challenges for the determination of the very moment a user’s turn is finished. This problem is particularly hard to solve when the user is allowed to provide input at any time during the interaction (*open-microphone* in contrast to *push-to-talk*). This leads to a conflict between flexibility and reaction time:

- *Interaction Patterns*—The user should be able to make use of every available input modality at any time and in any possible combination. As a consequence, the system must anticipate, when faced with some input event, that some other input event might

follow. However, due to the latencies introduced by the different recognition and analysis components, and due to the very diverse multimodal interaction patterns of users (see Oviatt et al. (1997); Oviatt (1999b)), the system has to wait some time after an input event before it can classify it as the last belonging to that turn. This means the longer the system waits for further input events, the more reliable the end of turn detection will be.

- *Reaction Time*—In order to reduce response time, the system should start processing as early as possible. Studies of human-human communication showed that humans are even able to predict the end of a turn before the last words are spoken. This means a system aiming at natural interaction should take as little time as possible to consider the turn finished.

Typically, the task of determining the end of a turn is done by the fusion component of a dialogue system. Today there are a number of approaches to determine the end of a turn of which we will summarize the most prominent ones:

- *Windowing*—A rather simple method where, after an input event has occurred, the fusion component waits for a specific time for further input. If no input event occurs, the current interpretation is sent to the dialogue manager, otherwise the time window is started over again. This method is employed, for example, in systems like the Match system (Johnston et al., 2001), or SmartKom.
- *Information Evaluation*—Here, the integration is performed after an input event has occurred. Then, the resulting interpretation is evaluated with regard to whether it fulfills some completeness requirements. If an interpretation is considered to be complete, it is passed on to the dialogue manager, otherwise a windowing technique is employed.
- Gupta (2003) proposes a method called *Dynamic Time Window* that permits the determination of the end of a turn with only small delays. This dynamic time window technique is based on the results of an empirical user study and uses a dynamic time window instead of a static one that adapts to the current input event based on previous learning.

4.3.5 Interaction Patterns and Input Synchronization

The temporal synchronization of the recognized input modalities is one particularly crucial aspect of a multimodal fusion component. Oviatt et al. call this process *multimodal synchronization* and have conducted experiments to identify common interaction patterns in humans. The results of (Oviatt et al., 1997) and of a later study (Oviatt et al., 2004) show that people do not utter multimodal commands at the same time but rather tend to utter them in a sequential manner. This means that any approach to multimodal fusion must be able to deal with slightly delayed input patterns.

This sequential interaction pattern appears especially in mobile scenarios where users have to hold the device near their mouth which clearly complicates entering accompanying gestures via the touch-screen. Wasinger et al. (2005) propose a set of temporal synchronization patterns for such a mobile scenario that takes the users' experience into account (see table 4.1). Based on the level of experience of a user the system adjusts the synchronization phase accordingly.

Temporal Relation	Duration
Before	Up to several minutes
During	4 to 5 seconds
After (standard)	500 - 750 msec
After (advanced)	1 sec
After (beginners)	+ 2 sec

Table 4.1: Input synchronization patterns used for a multimodal mobile shopping assistant (adapted from Wasinger et al. (2005)).

4.3.6 Summary

In this section we have discussed related work from the area of multimodal fusion. In general, all approaches are driven by the requirements of the specific application or available modalities and we are not aware of a generic approach that would also be able to deal with interactional contributions. The application of the unification operation (see section 5.2.4), has been identified as a common technique for dealing with the problem of semantic matching and integration. In general, we observed three major limitations that need to be addressed by a fusion component of a multimodal multiparty dialogue system:

- *Great variety of input modalities*—Nevertheless, most approaches are bound to their specific set of modalities so that any change would require changes in the code base.
- *Only limited reactive capabilities*—The discussed approaches show little or no processing capabilities for interactional discourse phenomena that require reactive actions. This means it would not be possible to realize a natural turn-taking behavior with these approaches.
- *End-of-turn detection*—Fast and reliable end-of-turn detection is crucial for a smooth exchange of turns. However, most of the discussed approaches make use of some kind of the windowing technique.

From these limitations, we identified the following general requirements on fusion that guided the development of our approach: (i) separation of application independent and specific processing rules and strategies, (ii) reduction of application specific processing rules and strategies, (iii) fast and reliable end-of-turn detection and (iv) the support for interactional contributions.

4.4 Approaches to Discourse Modeling and Processing

One of the earliest treatments of the resolution of pronouns and definite NPs in a running system is in the LUNAR system (see Woods (1978)). The anaphora resolution process within this system is based on a discourse model of *entities* that can be evoked through indefinite and definite NPs in a user’s query. The list of possible antecedents consists of the ten most recently evoked or referenced entities. The first entity that fits the anaphoric expression in a semantic fashion is taken to be the intended antecedent. This has the additional effect that

the antecedent is removed from its current position and inserted at the beginning of the list. This approach incorporates a discourse model that is only implicitly expressed by salience.

Wahlster defines a discourse model as “a knowledge source that contains the system’s description of the syntax, semantics, and pragmatics of a dialogue as it proceeds” (Wahlster, 1988, p. 102). Based on this notion of a discourse model, Wahlster defines the functions of a discourse modeling component of a dialogue system to (Wahlster, 1988, p. 102):

- Incrementally build up a discourse model.
- Store and update entities in it.
- Supply other components of the system with information about the structure and content of the previous segments of the dialogue.

In this section we will discuss related work from the area of computational approaches to discourse modeling and processing. But first we will briefly consider the structure of human-computer dialogues.

4.4.1 Discourse Structure in Human-Computer Dialogues

As pointed out in Ahrenberg et al. (1995), the language used during human-computer interactions differs from the language used in human-human communication. Moreover, Ahrenberg et al. also question the transferability of empirical results from one domain to another, e. g., from task-oriented dialogue to question-answering dialogue. In this respect Ahrenberg et al. follow a sub-language approach of Grishman and Kittredge (1986). But Ahrenberg et al. (1995) go one step further in that they argue for computational theories of discourse that should focus on the computers’ processing and not on general theories of language use. In what follows, Ahrenberg et al. (1995) claim that the local context of a segment is given by the immediately preceding segment.

For question-answering dialogues, the situation is somewhat different. Chai and Jin (2004) highlight the importance of a semantically rich discourse representation that supports the resolution of context dependent phenomena (Chai and Jin, 2004). Bertomeu et al. further argue that the original claim for only considering the immediately preceding segment cannot be transferred to the domain of question-answering dialogues.

4.4.2 Contextual Enrichment

During the course of a dialogue, each new user contribution has to be interpreted in light of the previous discourse context and in particular of the previous utterance of the same speaker. Sometimes, it is suitable to incorporate compatible information into the representation of the new contribution—a process that is called contextual *enrichment* (Alexandersson and Pflieger, 2006; Pflieger et al., 2003a). Consider, for example, the following dialogue fragment (from (Alexandersson and Pflieger, 2006, p. 238)):

- (71) **User:** “*What’s playing at the movies tonight?*”
- (72) **System:** [Displays a list of movies] “*Here ↗ you can see tonight’s movie theater program.*”
- (73) **User:** “*And what’s on TV?*”

(74) **System:** [Displays a list of broadcasts] *“Here ↗ are tonight’s broadcasts.”*

(75) **User:** *“Is there a movie with Arnold Schwarzenegger?”*

Turn (73) and (75) are examples of where we need access to the previous discourse context to obtain the correct interpretation. In (73), the user asks for tonight’s television program and not for the one that is currently running. Even in (75), the user still incorporates this implicit temporal restriction into the new request. As discussed in (Alexandersson and Pfeleger, 2006; Pfeleger et al., 2003a,b), contextual enrichment can be realized by means of overlay which is a default unification-like operation (see Alexandersson and Becker (2001, 2003); Alexandersson et al. (2006); see also 5.2.4). The basic idea is that the semantic representation of the new contribution is overlaid with the previous contribution of the same speaker in order to incorporate all information that is not overwritten by the new contribution.

However, it is not always reasonable to incorporate contextual information into the interpretation of the new contributions. As Streit and Krieger (2004) note, the application of default unification without additional control mechanism shows the tendency to over-accumulate information. Consider the following dialogue fragment as an example:

(76) **User:** *“What’s playing at the movies in Saarbrücken?”*

(77) **System:** ...

(78) **User:** *“And tomorrow?”*

(79) **System:** ...

(80) **User:** *“What’s playing at the movies in Kaiserslautern?”*

While the transfer of contextual information makes perfect sense for turn (78), it is counterproductive for the subsequent user contribution in turn (80). Besides this, whether it is suitable to incorporate contextual information also depends on the type of dialogue and who is having the initiative. In information-seeking dialogues, for example, it makes sense in situations where the user has the initiative. In task-oriented dialogues when the system has the initiative, a transfer of contextual information seems less suitable. Usually, speakers also provide clues as to whether or not their contributions are supposed to be enriched with information stemming from the previous contribution (see for example the elliptical expression in 78).

Furthermore, it remains unclear to what extent this approach that was originally developed for dyadic dialogue can be used for multiparty interactions. In the original version of the approach, the enrichment of a new contribution has always been done with respect to the previous utterance of the same speaker. In multiparty discourse, however, it is unclear whether it is suitable to perform contextual enrichments of contributions where the speaker changed, as in the following example:

(81) **Speaker A:** *“What’s on TV tomorrow night?”*

(82) **System:** ...

(83) **Speaker B:** *“What movies are playing in the theater?”*

Here, B’s question can either be interpreted based on the contextual information set up by A’s utterance or on its own depending on the situation. If the two participants appear as a group with a joint goal, it would be suitable to interpret A’s contributions in the light of B’s. Otherwise, there is a preference for the context-independent interpretation.

4.4.3 Computational Approaches to Discourse Modeling

Grosz and Sidner—Attention, Intentions and the Structure of Discourse

The centering theory of Grosz and Sidner (1986); Grosz et al. (1995) introduced a contextual representation that assigns a unique *backward-looking center* C_b and a ranked list of *forward-looking centers* C_f to each utterance in a discourse. The elements of this C_f list are the possible candidates for antecedents of the next utterance. There is no numerical weight attached to the elements of the list, they are simply ordered relatively to each other. Many approaches for anaphora resolution deal with the centering theory (see Hahn and Strube (1997)). In addition to that, centering theory and earlier focus-based models have also been adopted for the generation of text as for example in the work of Dale (see Dale (1992)).

A Three-tiered Discourse Model

In her PhD-thesis, Luperfoy proposes a three-tiered discourse representation that is capable of representing context-dependent NPs (Luperfoy, 1991). This discourse representation is based on three abstract layers: (i) the linguistic analysis of the utterance, (ii) the knowledge base or belief system, the interpreter's static theory of the world of reference, (iii) the discourse model representing the information content of the discourse. She defines three classes of objects that appear on three different layers of the discourse representation:

- 1 *Linguistic Objects (LOs)* which encode the linguistic representation of *referring expressions*⁸. This linguistic representation includes surface information (like the word or phrase uttered or typed) as well as abstract representations of that expression (like phonological, morphological, syntactic and lexical semantic representations). For each occurrence of a referring expression in the utterance, an LO is introduced into the linguistic representation.
- 2 *Belief Objects* which are instances of a belief system (classes and individuals and the relations between them). The belief system is used as a knowledge base and provides terminological and assertional information. Terminological information consists of information about the ontological type hierarchy, properties of kinds and default properties of their non-specific instances (including rules relating kinds, sub-kinds and default instances of kinds). Assertional information is particular information about individuals believed to exist in the real world.
- 3 *Discourse Pegs* which represent instances that have been introduced into the discourse. There is only one peg for each concept represented in the discourse, regardless of how many NPs occur referencing that concept. Additionally, a peg has access to the linguistic information of all LOs mentioning it, but does not share those linguistic features with its LOs. Each peg has three classes of information: linguistic, discourse and belief information.

The diagram in figure 4.15 labels the relationship between the three levels of the discourse representation. The *is-sponsored-by* relation relates anaphoric LOs to their antecedents and the *is-anchored-by* relation relates an LO to a peg. Co-anchoring reflects the configuration

⁸Luperfoy terms every object of the discourse that refers to an entity in the real world as a *referring expression*. A referring expression can be viewed as the mentioning of a concept of the real world.

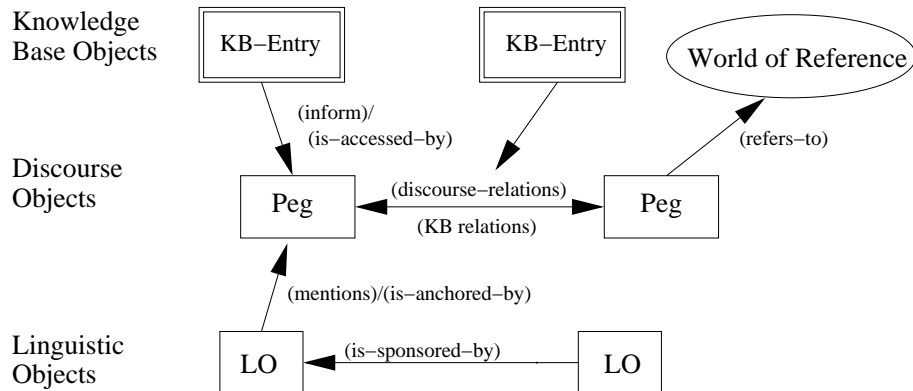


Figure 4.15: The Three-Tiered Discourse Representation: Orientation of Discourse Pegs to other objects (adapted from Luperfoy (1991)).

when two or more LOs mention a common peg, independently from whether they stand in any linguistic sponsorship relation to each other or not.

For the interpretation of anaphoric NPs, it is useful to differentiate between linguistic information of linguistic objects and discourse information within discourse pegs. Linguistic objects can, for example, have a grammatical gender, and in contrast to that, pegs can have a natural (semantic) gender.⁹ As Luperfoy points out, even in languages with semantically arbitrary (grammatical) gender such as German, this distinction plays a role in anaphora resolution. In the following, a (simplified) example of Luperfoy will be reproduced for two reasons: (i) to illustrate the basic mechanisms of Luperfoy’s approach and (ii) to stress the importance of the distinction between grammatical and semantic gender.

Luperfoy (1992) extends this three-tiered approach to multimodal dialogues. In general, this separation of objects at the linguistic and discourse level provides a flexible framework for dealing with the types of referring expressions introduced in section 3.2.3. But Luperfoy’s approach lacks two key features since it neither incorporates VP-expressions nor the physical context.

Domains of Reference

Salmon-Alt (2000) introduces a discourse model based on the notion of *domains of reference*. These domains represent the set of available referents, i. e., discourse entities that have either been introduced during the previous discourse or through visual perception. However, a domain of reference does not necessarily comprise all available entities, but rather represents different recollections of these entities. Different types of referring expressions (indefinite, definite or pronoun) select differently structured domains and have different consequences for the restructuring of their domain. Discourse and perceptive referents are represented in terms of mental representations (MRs).

⁹Gender serves not only to classify entities into broad classes based on their meanings (human/nonhuman, male/female), but serves various grammatical functions as well. These grammatical functions are the reasons for a gender system that does not reflect the natural (semantic) gender. In German, names of animate beings are masculine or feminine according to their sex. Young people, however, are neuter. We therefore have to differentiate between a grammatical and a natural (semantic) gender.

An MR (see for example figure 4.16) consists of attribute-value pairs including a unique identifier (in this example @P1&P2), a type (in this example |pyramid|), and a partition which represents compositional information by pointers to other MRs. Since an MR may represent both individual objects and collections of objects, an MR may contain zero, one or more partitions. Having more than one partition represents a membership of a specific MR in more than one group. Every partition has a differentiation criterion (DC), for example in the case of a visually presented object this would be its horizontal position on the screen. Each element of the partition represents a sub-component (for example @P1) identifiable explicitly by the value of its differentiation criterion ($v(DC)$). No more than one element of a partition may be in focus, according to perceptive or discourse based salience.

@P1 & P2	
Type	= pyramid
Partition	= part_1
DC	= horiz_position
v(DC) = left	v(DC)=right
@P1	@P2

Figure 4.16: Example of a mental representation in the approach of Salmon-Alt (2000).

In this framework, the discourse history consists of the domains of reference (DR) that have been made available over the course of processing. DRs are either individual entities (represented by one MR) or sets of entities (represented by a set of MRs). Sets of entities are created via a grouping operation which may be caused by linguistic information (coordination or enumeration) or perceptive information (similarity or proximity). Thus, two or more existing MRs are linked to a newly created MR as members of a single partition.

The Context Resolution Server of the Galaxy System

(Filisko and Seneff, 2003) present a context resolution algorithm developed for the *Galaxy Communicator* architecture (Seneff et al., 1998). The Galaxy system is basically a central hub which establishes means of communication among a set of specialized language processing servers (e. g., speech recognition, dialogue management, or context resolution). Among other tasks, the context resolution server of the Galaxy system is responsible for the resolution of anaphoric, deictic and elliptic expressions.

The central context representation of the context resolution server is the *history record* which stores the semantic frame of each utterance. Additionally, there is a temporally ordered *discourse entity list* comprising every topic mentioned in the dialogue. Mouse-based click-gestures the user can elicit via a web-browser based interface are also stored in this discourse entity list. These click-gestures take priority so that they will always be added to the top of the entity list. However, there seems to be no further structuring of this context representation besides temporal salience.

The resolution of referring expressions such as anaphora or deictic expressions is handled in a straightforward manner: If faced with such an expression, the context resolution server searches the list backwards until it finds a matching discourse entity. What exactly defines a *match* is not explained in Filisko and Seneff (2003). Elliptical expressions are resolved with the semantic frame of the previous utterance.

The Multimodal Discourse Modeler of the SmartKom System

The multimodal discourse modeler (DiM) of the SMARTKOM system (Wahlster, 2003a,b, 2006) has set the basic roots for our present work. The complete approach is described in (Alexandersson and Pflieger, 2006; Pflieger et al., 2003a,b; Pflieger, 2002). Here, we will only give a brief overview of the underlying context representation¹⁰.

The approach to discourse representation employed in DiM is based on that of Luperfoy (1991) and Salmon-Alt (2000). It extends the three-tiered context representation of Luperfoy (1991) by generalizing her linguistic layer to a *modality* layer (see Figure 4.17). Additionally, we have adopted some ideas from Salmon-Alt (2000) by explicitly representing compositional information of discourse objects. The advantage of this approach to discourse representation lies in the *unified* representation of discourse objects introduced by the different modalities.

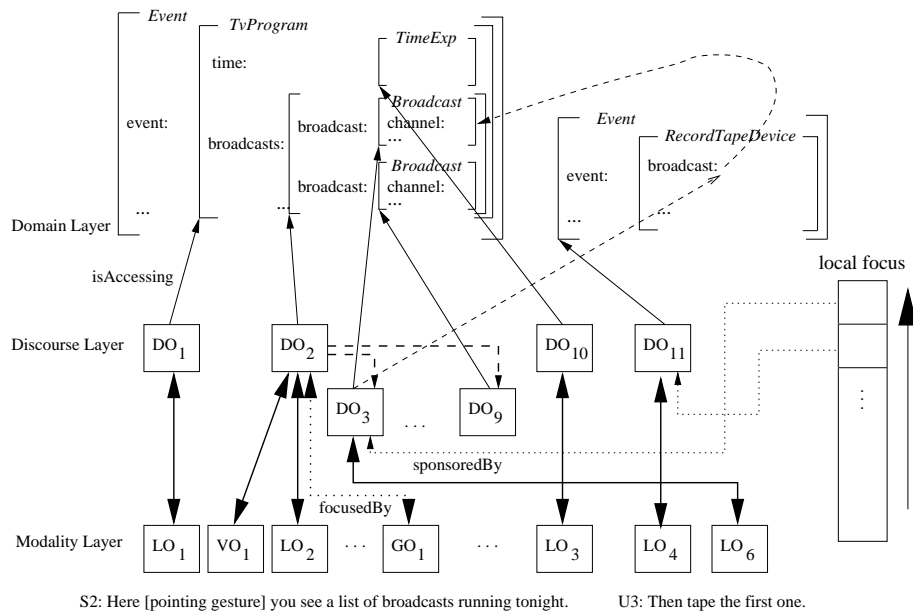


Figure 4.17: The multimodal context representation of SmartKom: The dashed arrow indicates that the value of the broadcast in the (new) structure on the right is shared with that of the old one (on the left).

Modality Layer An object at the modality layer (henceforth MO) encapsulates information about the concrete realization of a referential object depending on the modality of presentation. Corresponding to the three different types of presentation, the modality layer is an accommodation of the different types of objects introduced by the different modalities:

¹⁰Most of this material can be found in Pflieger et al. (2003a).

- *Linguistic Objects* (LOs): For each occurrence of a referring expression in a generated or interpreted utterance, one LO is added (Luperfoy, 1991).
- *Visual Objects* (VOs): For each visual presentation of an object that can be referred to, one VO is added.
- *Gesture Objects* (GOs): For each gesture performed either by the user or the system, one GO is added.

Each modality object is linked to a corresponding discourse object.

Discourse Object Layer The central layer of the discourse model is the discourse object layer. There, a discourse object (DO) represents a concept which potentially serves as a referent for referring expressions including objects, events, states and collections of objects. Every time a concept is newly introduced into discourse by speech, a DO is created. A DO is also created for directly perceived concepts, e. g., graphical presentations (Salmon-Alt, 2000).

Each DO relies on two classes of information, (i) modality specific information and (ii) domain information. MOs at the modality layer *mention* a DO but are only able to modify the DO with respect to the domain model by adding new information to the corresponding domain representation. For example, an LO with syntactic gender-marking *female* constrains the linguistic relations it enters but does not affect its corresponding DO. Note that DOs are unique at the discourse layer. For each concept introduced during discourse, there is only one DO, regardless of how many MOs mention this concept. The domain information of a DO represents the current information state of a DO in terms of the domain model. For each mention of a DO, there is at the domain object layer, a corresponding application object or sub-object. To provide access to these instances of the domain model, the domain information consists of a list of pointers to these instances. Additionally, the domain information includes a unified representation of these instances that is used for the identification of identical DOs and for the resolution of referring expressions.

The compositional information of DOs representing collections of objects is provided by partitions (Salmon-Alt, 2000). Such partitions are based either on perceptive information (e. g., the set of movies visible on the screen), or discourse information (e. g., “*Do you have more information about the first and the second movie?*” in the context of a set of movies presented on the screen). Each element of a partition is a pointer to another DO, representing a member of the collection. The elements of a partition are distinguishable from one another by at least one *differentiation criterion* ($v(DC)$) like relative position on the screen, position within a set, size or color. Within a partition, one element at most may be in focus, according to gestural or linguistic salience.

Domain Object Layer The domain object layer provides the mapping between a DO and instances of the domain model. Instances of the domain model provide a semantic representation of actions, processes and objects. In SMARTKOM, our domain model is described in OilEd¹¹ and provides a type hierarchy.

¹¹OilEd is a free ontology editor supporting DAML+OIL, developed by the University of Manchester; see <http://oiled.man.ac.uk/>.

GALATEA: A Discourse Modeler for Conversational Spoken Language

The GALATEA discourse modeler (Skantze, 2006) was developed for the HIGGINS project (Edlund et al., 2004), which is a testbed for developing error handling strategies for spoken dialogue systems. The domain of the HIGGINS system is a monomodal (speech input and output only) pedestrian city navigation and guidance system where the system has no access to the exact location of the user (there is no GPS information available). Thus, the system has to compute the users' position from verbal descriptions of the surroundings.

The discourse modeler of the HIGGINS system has to fulfill three main tasks: (i) the resolution of ellipses, (ii) the resolution of anaphora and (iii) the maintenance of grounding information. The component is implemented in Oz¹². Elliptical expressions are resolved by transferring them into full propositions by means of so-called *context rules*. These rules specify how a given elliptical expression has to be integrated into a given conversational act. The anaphora resolution algorithm of GALATEA is realized as a rather simple top-down search for a semantically compatible (i. e., unifiable) antecedent where new or un-resolvable objects are put on top of the candidate list. Moreover, the anaphora resolution of GALATEA has no access to the domain database and thus there is no mapping of referring expressions to real world objects.

GALETEA also tracks the grounding status of concepts, i. e., information about who added at what time the concept to the model and how confident the system is about the concept. This grounding information is used for error handling by the action manager. The actual grounding status of an object is determined by the following rather simple distinction: "If a concept is only mentioned by the user with a low confidence score, it has a low grounding status. If it has been mentioned by the system and/or by the user with high confidence, it has a high grounding status" (Skantze, 2006, p. 185).

Even though the error handling strategies of the HIGGINS system seem to be very elaborate, the discourse modeler is based on a rather simple stack-based representation of the ongoing dialogue (similar to the approach of Woods (1978)). This might be sufficient for the task at hand but is most likely not applicable for other types of referring expressions like deictic references to the physical context. It is also questionable whether the total neglect of syntactical information will be sufficient for the resolution of pronouns and the like.

Contextual Reasoning Using Lambda Calculus

Boye et al. (2004) present an approach to reference resolution that employs a single machinery based on lambda calculus to resolve both references to the previous dialogue and the visual context. Their approach follows the tradition of recency-based resolution techniques (e. g., Mitkov (1998)). Boye et al. (2004) go beyond simple grammatical restrictions as they employ a type-based semantic compatibility constraint (i. e., they look for the most recent object with compatible type). However, even though the authors highlight this compatibility approach, it again boils down to the same as in early work of Woods (1978).

The actual context model of Boye et al. comprises three distinct areas:

- 1 A *visual context history* consisting of a recency-ordered list of sets of objects, each of which represent an individual visual context (i. e., the objects displayed on the screen at a specific point in time). Every time the appearance of the display changes, a new visual context will be added to the list.

¹²Mozart Oz is a programming language for distributed applications, see <http://www.mozart-oz.org>.

- 2 A *dialogue history* which is a recency ordered list of typed combinators. A typed combinator represents a resolved utterance of the system or the user. The primary purpose of the dialogue history is the resolution of pronouns and ellipses.
- 3 An *event history* that supports the resolution of references to previous actions.

Discourse Modeling in the CEDERIC System

The CEDERIC (*Case-base Enabled Dialogue Extension for Robotic Interaction Control*) is a dialogue system for the interaction between a human and a robot, in particular the WITAS unmanned Yamaha RMAX helicopter (see Doherty et al. (2000)). The discourse model of CEDERIC as described in Eliasson (2005), is based on an early version of our discourse model (see Pflieger et al. (2003a)).

Their approach extends our work by integrating it into a case-based reasoning (CBR) system. The basic components of the CEDERIC system are a *case base*, a *domain knowledge*, a *discourse model* and a *case base manager*. The domain knowledge is modeled by means of an ontology that covers a representation of the world (e. g., buildings, their attributes like colors or their locations). The discourse model basically uses the same data structures as in our approach. The cases of the CBR system also comprise discourse information. This part of the case states what the global focus and the focused discourse object should look like. Thus, it defines the contextual constraints of a case. In total, a case in this approach consists of five different parts:

- 1 *The problem* which comprises the words and classification according to the domain knowledge.
- 2 *The discourse information* which describes what the currently active global focus space and its discourse objects should look like.
- 3 *The update according to the problem* which defines how the discourse model has to be updated with new information.
- 4 *The solution* which contains the reaction of the system.
- 5 *The update according to the solution* which defines how the discourse model has to be updated when the solution has been executed.

Case matching takes place as follows: When a new utterance from the user or a message from the robot enters the system, it will first be analyzed with respect to the grammar. Then the case base is searched for cases that show both matching utterances at the problem description and discourse information that are compatible with the currently active focus space. If no case matches, CEDERIC assumes a that the current utterance belongs to a previous, unfinished dialogue and tries to find a compatible case by by traversing the global focus stack.

Cross-Model Reference Resolution

One of the first treatments of cross-modal references in a dialogue system was realized in the SHRDLU system of Winograd (1972). Here, the user has the possibility to refer to objects displayed on the screen using linguistic expressions like “*pick up the red box.*” In order to

be able to resolve such a referring expression, a virtual agent needs a spatial representation of the physical environment. Besides the spatial organization, this physical representation must encompass salient information like color, shape and size and also the current point of view. (Maass, 1996; Gapp, 1996; Blocher, 1999), for example, put forward comprehensive models of the physical surroundings that are used for the incremental generation of direction instructions.

Byron et al. (2005) introduce a reduced model for visual attention that supports the process of resolving cross-modal references. They base their model on an experimental study where two subjects cooperated to find hidden treasures in a first-person graphical world (rendered by the QuakeII engine¹³). The partners communicated through headset-mounted microphones and the audio recordings of the sessions were transcribed. In addition, each player's activity in the virtual world was taped. Based on the data collected with this experiment, Byron et al. (2005) developed an algorithm for modeling visually salient entities. The algorithm basically quantifies the novelty of an object and how it changes over time using a *Uniqueness* (U) parameter. Moreover, an object must not to be in the field of vision, as the algorithm also considers the amount of time since it was last seen (*Recency* (R)) and how long it was present before it vanished (*Persistence* (P)).

A major objection to Byron et al. (2005) is that they have not yet integrated their model and algorithms into a running system so it is hard to tell whether they will really be able to achieve the performance of their *paper-and-pencil* evaluation. However, because of the promising initial result, we have taken up their model and integrated it into our context model. Thus, the algorithm for computing visual attention will be taken up in Chapter 8.

Resolution of Indirect Anaphora

There have been a number of computational approaches put forward for the resolution of indirect anaphora or bridging references, of which here we will briefly discuss a small selection. Grosz used a semantic network consisting of domain objects as the basis for her focus model (Grosz, 1977). The model comprises two distinct focus representations: (i) an explicit focus and (ii) an implicit focus. The difference between the two focus representations is that the items in the first focus explicitly participated in the preceding discourse, and the items of the implicit focus are only connected to the items in the explicit focus. Grosz makes use of two concepts called spaces and *vistas* to partition the network into smaller groups of entities. Grosz uses hand-coded focus spaces to group entities that should appear together in the same focus space. Vistas, in turn, are used to encode association relationships between entities.

Bunescu (2003) present a web-based machine learning approach. This approach is based on the idea of searching the web for a particular type of lexico-syntactic patterns. Using statistics on these patterns, they attempt to recover the antecedents for a subset of two types of anaphoric relations: identity anaphora and associative anaphora. Bunescu (2003) discuss preliminary results that show a performance of the algorithm that is in the range of other approaches. Fan et al. (2005) introduce a slightly different WordNet based approach which they call *Semantic Path Search*. The idea is to view the problem of resolving indirect anaphora as the problem of how to use background knowledge to infer relations among linguistic constituents. This approach makes use of what they call an *interpreter* that tries to find relations between pairs of concepts. The basic algorithm goes as follows: First, the referring expressions and the antecedent candidate are mapped to nodes (C_1 and C_2) in the knowledge base

¹³See <http://www.id.com>.

(which contains all data from WordNet). Then, two breadth-first search processes traverse the relations of the knowledge base. One search process starts at C_1 and looks for C_2 or any superclass or subclass of it, and the other one searches the other way around. Eventually, a sorting function ranks the interpretations that have been found. Fan et al. (2005) present two experiments that show slightly improved precision and clearly improved recall rates compared to other approaches. However, since their knowledge base contains only the WordNet data, they cannot account for indirect references like event—role or cause—consequence. Moreover, they consider a fixed window of six sentences for candidate referents ordered by salience. So, they do not account for any other contextual effects like, for example, accumulation of activation.

4.4.4 Summary

In this section we discussed a selected number of computational approaches to discourse modeling. All these approaches are tailored for the task of reference resolution. The individual underlying models of the discourse, however, differ significantly, and so do the number of supported types of referring expressions.

Viewed on an abstract level, reference resolution breaks down into three sub-tasks: (i) compute the list of possible referents, (ii) order the list according to the number of salience factors and (iii) select the first matching referent. A major shortcoming of the discourse models reviewed in this chapter is that most of them have no integrated long-term memory and thus model only explicitly referenced entities. However, many definite references to objects are actually references to existing knowledge and will never be introduced to the discourse before actually being used. The interpretation of these references requires access to a long-term memory representing all those facts and events that are typically available to humans. Moreover, to be able to participate in natural dialogues, participants must have a good idea of what the other participants know and what they do not. Consider, for example, the following discourse fragments:

(84) **Mary:** “Do you want to take the car, or can I take it?”

(85) **Peter:** “Well, I met the guy from CNN yesterday, you know...”

Paul: “Great, what did he say?”

In fragment (84), Mary presupposes that the addressee of her utterance most probably will be able to infer which car she means. This means there must be some distinct car that is prominent for both of them so that Mary does not need to further explain which car she means. If, for example, Mary and the addressee own that car together, it is in most situations clear for both of them what car is meant by “the car.” In fragment (85), the situation is slightly different. Again, the first speaker uses a definite reference to some guy without introducing him beforehand. However, in this example the speaker seems to be unsure whether the addressee will be able to understand the reference and therefore adds some more information that is supposed to help to narrow down the intended referent. The addressee in this example is perfectly able to resolve the reference and directly refers to “that guy” using the personal pronoun *he*, a clear indication that he is absolutely sure that he knows who the other person is talking about.

Another problem with the existing discourse models is that they focus on the representation of standard information like a syntactic and semantic description of the contributions, a set of referenced discourse entities, an attentional structure and anaphoric links. However, as

Wahlster (1988) highlights, there are other ingredients that are required for a comprehensive discourse representation.

4.5 Multiparty Human-Computer Interactions

In contrast to dyadic interactions, multiparty human-computer interactions require the virtual agents to be able to participate in turn-taking management and to identify whether they are addressed by a perceived utterance. In this section we will briefly discuss related work in these areas.

4.5.1 Computational Approaches to Turn-Taking Management

As discussed in section 3.3.2, when more than three people engage in conversation they have to organize their contributions in order to avoid constantly overlapping speech. Today's dialogue systems mostly do not incorporate special means for dealing with the organization of the turn-taking protocol. The reason is that most systems focus on dyadic interactions, which reduces the question of when to speak to the question of whether the user is currently speaking. However, multiparty dialogue systems require additional processes in order to achieve a smooth exchange of the floor.

Thórisson (1996, 2002) introduced an architecture for a reactive multimodal dialogue system that incorporates a predictive model of turn-taking (see also section 4.1.1 for an overview of Thórisson's approach). The key to this approach is a layered processing with different priorities consisting of (i) a *content layer* with low priority, (ii) a *process control layer* with medium priority and (iii) a *reactive layer* with high priority. Perception is done via a collection of Perceptor modules which take in sensory data, or partially processed data from other Perceptors, and compute further results. Perceptors further decompose into Unimodal Perceptors and Multimodal Integrators which integrate the output of the Unimodal Perceptors. The output of the Perceptor modules is read via shared blackboards by so-called Deciders. Decider modules either keep track of the dialogue state and turns (Covert State Deciders) or make decisions about an agent's visible behavior (Overt Deciders). The Perceptors and Deciders are responsible for the turn-taking system. The Multimodal Integrators in particular are responsible for the interpretation of the turn-taking behavior of the other participant; consider, for example, the following representation of a Multimodal Integrator:¹⁴

```
Other-is-giving-turn
ACTIVE-DURING-STATE: Other-Has-Turn
CONDITIONS:
(AND
  (Other-is-speaking = F)
  (Or
    (AND
      (Other-is-looking-at-me = T)
      (Other-is-facing-me = T))
    (AND
      (Other-is-looking-at-me = T)
      (Other-is-gesturing = F))
    (AND
      (Other-is-gesturing = F)
      (Other-is-facing-me = T))))
```

¹⁴Thórisson (2002) uses a Lisp-like syntax for representing processing rules.

This rule identifies whether the other dialogue participant is giving the turn to the virtual agent and it is applicable if one of the three conjunctions in the disjunction is applicable. The actual turn-taking behavior of the virtual agent is then triggered by means of Overt Decision modules in the reactive layer; consider for instance the following rule:

```
Show-Im-taking-turn
EL: 5000 msec
BehaviorRequest: Show-Im-taking-turn
FIRE-CONDS: (I-take-turn = T)
RESTORE-CONDS: (I-take-turn = F)
```

This approach realizes a detailed implementation of the turn-taking system. However, it is focused on dyadic interactions and it remains unclear whether it is feasible to apply it to multiparty interactions. Perceived turn-taking signals are not the source of information that influences the turn-taking system. Following Schmitt (2005), the management of the turn-taking system can also be influenced by the fact that people sometimes have the urge to speak because of group-dynamic processes. Schmitt computes, based on a *turn-taking activation level*, the current urge to speak for virtual agents. In this approach, the agent with the highest activation is supposed to take the turn. This approach, however, has been applied in a system that simulates conversations between multiple virtual characters. Whether it can be applied to multiparty conversations between humans and virtual agents still needs to be investigated.

4.5.2 Addressee Identification in Dialogue Systems

Until now, the identification of the intended addressee of a contribution has not received much attention in classic multimodal dialogue systems, since most systems only deal with a single user and a single artificial counterpart so that there is simply no need to identify the intended addressee. However, with the advent of multiparty dialogue systems like multimodal meeting managers (e. g., the AMI project¹⁵) or multiparty virtual reality systems (e. g., the Mission Rehearsal System (see Hill et al. (2003)), or VIRTUALHUMAN (see Reithinger et al. (2006))), the need for a robust and fast addressee identification came to the front. First attempts to model the relevant features and inference mechanisms for automatic addressee recognition have been put forward by Jovanovic and op den Akker (2004) and Traum (2004) which will be discussed in subsection 4.5.2 and 4.5.2, respectively. But there is an important difference between these two approaches: The MRE system has to perform the addressee identification online while the user is interacting with the virtual characters. In the AMI project, however, the addressee identification is performed on recordings of the meetings which means that this approach does not have to deal with real-time demands.

Addressee Recognition in a Meeting Manager

Jovanovic and op den Akker put forward a rule-based approach to automatic addressee detection for small group discussion meetings (Jovanovic and op den Akker, 2004). They base their approach on four basic categories of information that support addressee recognition: (i) speech, (ii) gaze direction, (iii) gesture and (iv) context. The following will provide a brief summary of these informational sources: (i) *Linguistic markers*, (ii) *Name detection*, (iii)

¹⁵See <http://www.amiproject.org/ami-scientific-portal/ami-overview>.

Dialogue acts, (iv) *Gaze direction*, (v) *Gestures* and (vi) *Context*. With respect to the context Jovanovic and op den Akker distinguish four aspects: (i) interaction history comprising both conversation and nonverbal events, (ii) meeting action history and the series of previous meeting actions including the current one, (iii) user context comprising participants' names, gender, social roles (status roles and closeness), (iv) spatial context comprising participants' location, locations of the environmental objects, distance between participants, participants' visible area.

The actual algorithm that Jovanovic and op den Akker (2004) propose comprises a layered automatic addressee detection that is based on the observation that participants can only predict the most probable addressee of an utterance. Their solution is rule-based and operates on candidate lists: The first processing layer processes information obtained from the verbal part of the utterance and returns a list of possible addressees with corresponding probabilities. In the subsequent steps, the algorithm integrates other sources of information like gaze behavior and gestures.

In a subsequent paper, Jovanovic et al. (2006) develop and test a different approach that is based on a Bayesian Network and Naïve Bayes classifiers. In particular Jovanovic et al. (2006) focus on the question of to what extent the meeting context can aid the classification performances. Their results show that the classifiers perform best when conversational context and utterance features are combined with information about the speaker's gaze. Information about the meeting context, however, showed only little improvement in performance.

Addressee Recognition in the MRE System

Traum describes an approach to addressee recognition (see 4.18; see Traum (2004)) that has been used within the *Mission Rehearsal Exercise* (MRE) system (Traum and Rickel, 2002). Traum's approach, however, is based on a less elaborate set of informational sources than the one of Jovanovic and op den Akker (2004). Basically, he distinguishes the following types of information:

- *Vocative expressions*—the speaker directly indicates the addressee through calling by name or role.
- *Content of the utterance*—some content already determines who is addressed (if, for example, nobody else knows the answer).
- *Context*—the previous speakers or addressees.
- *Gaze or body orientation*—the individual the speaker is looking at.
- *Gesturing*—attention getting and deictic gestures.

Traum emphasizes that some of the information types require advanced multimodal input facilities that might not always be available in current multimodal dialogue systems.

4.6 Conclusion and Discussion

In this chapter we have presented an overview of related existing technology in the area of multimodal fusion, context modeling and discourse processing. Most of the approaches are tailored to the use in dyadic multimodal dialogue systems, except for those approaches


```

If utterance specifies addressee (e.g., a vocative or utterance of just a name when not
expecting a short answer or clarification of type person)
then Addressee = specified addressee

else if speaker of current utterance is the same as the speaker of the immediately
previous utterance
then Addressee = previous addressee

else if previous speaker is different from current speaker
then Addressee = previous speaker

else if unique other conversational participant
then Addressee = participant

else addressee unknown

```

Figure 4.18: The MRE addressee identification algorithm (adapted from Traum (2004)).

that originate from the MRE or the AMI project. Furthermore, there is to our knowledge no approach covering both multimodal fusion and discourse processing in an integrated way. The major shortcoming of the context and discourse models discussed in this chapter is that they do not incorporate all relevant information that is needed to get a complete representation of an ongoing multiparty interaction. Our context model, which is introduced in Chapter 6, seizes on the shortcomings of these models and incorporates the entire conversational state.

With respect to the approaches to multimodal fusion, a strong focus on the integration of spoken utterances with deictic pointing gestures or sometimes handwriting input can be observed. None of the reviewed models considers other nonverbal behavior like gaze behavior or other types of interactional contributions. Moreover, the discussed approaches to multimodal fusion do not incorporate contextual information that goes beyond the current turn in to the integration process. This means that these approaches cannot deal with ambiguous monomodal input that requires contextual information about the current conversational state. As discussed in section 3.4, this has the consequence that these approaches are not able to deal with interactional contributions that function as signals for the regulation of the turn-taking system.

Based on this review and the discussion of the context-dependent discourse phenomena in Chapter 3, we are now able to define the central tasks and functionality for our integrated multimodal discourse interpretation component:

Multimodal Interpretation and Integration (Fusion): Processing incoming monomodal events on both the propositional and the interactional level.

Context Modeling: Maintaining a comprehensive model of the context of the interaction (including the conversational setting as well as interactional and propositional aspects of the discourse).

Reference Resolution: Supporting the full range of referring expressions such as anaphora, exophoric and deictic expressions.

Resolution of Elliptical Expressions: Supporting the full range of elliptical expressions.

End-Of-Turn Detection: Supporting fast and reliable end-of-turn detection. This is the primary key for a smooth exchanges of the speaking turn.

Addressee Identification: Supporting the identification of the intended addressee by means of verbal, nonverbal and contextual factors.

Triggering Reactive Feedback: The ability to output reactive feedback such as backchannel feedback or turn-taking signals.

These tasks will be covered by the integrated multimodal fusion and discourse processing component that will be presented in Chapter 8.

Chapter 5

Representing Meaning in Multimodal Dialogue Systems

Choosing a well suited format and internal model for representing meaning in multimodal dialogue systems is a key aspect in the design process of such applications. Since one of the overall goals of this work is to develop a generic and reusable component for multimodal fusion and discourse processing, a lot of effort has been put into this question. In this chapter we will discuss the foundations of a number of recent languages and formats for representing meaning and metadata in multimodal dialogue systems. In section 5.1 we will give a brief introduction to the topic before we will discuss in section 5.2 existing notations and languages for representing meaning. Section 5.3 discusses two representation languages for modeling multimodal user input, while section 5.4 gives a brief introduction to ontology-based representation of meaning.

5.1 Introduction

An important aspect in the process of designing and developing multimodal dialogue systems is the specification of a representation language that is capable of representing both the form and the content of verbal and nonverbal contributions. Such a representation language can be employed as a uniform communication interface between the internal modules of a dialogue system. As Gurevych et al. (2003b) argue, a uniform representation language (i. e., an ontology) is of particular importance for the development of such systems, as it reduces the conversion of data that is exchanged between the individual modules and it permits the reuse of existing modules for new dialogue systems. A common approach in recent multimodal dialogue systems (e. g., SmartKom, COMIC, SmartWeb) is to use a representation language such as M3L or EMMA (see section 5.3) for modeling meta information (e, g., used modalities of expression, time stamps) in combination with a structured, ontology-based representation of meaning and real world objects.

The concept of an ontology originates from philosophy and is concerned with “what is, the kinds and structures of objects, properties, events, processes and relations in every area of reality” (Smith, 2003, p. 155). Smith highlights that ontologies have gained a lot of interest in computer and information science. Ontologies are often viewed as being quite similar to semantic networks. They are, however, considered to employ richer semantic relationships among terms and attributes. Today, ontologies are used in various areas of computer science

but in particular in artificial intelligence. Here, ontologies are considered to consist of the following aspects:

Classes: The taxonomy of an ontology consists of classes (also called *concepts*) representing groups or sets of objects that share several features (i. e., attributes and relations; see below). An example of a class would be *Animal*, the class representing all classes of animals.

Attributes: Each object in an ontology can be assigned attributes consisting of a *name* and a *value*. These attributes are used to store information that is specific to the objects it is attached to (e. g, the first name or last name of a person).

Relations: Relations are attributes that express connections between objects of the ontology. The most prominent example is the *is-a* or *is-subtypeOf* relation that is used to define taxonomies.

Individuals: *Individuals* or *instances* are the actual instantiations of classes. An object that represents a particular person (e. g., the football player Michael Ballack) is realized as an individual that is an instance of the class *FootballPlayer*.

In artificial intelligence, an ontology is viewed as “a specification of a conceptualization” Gruber (1993). This means it is a description of the concepts and relationships that exist for an agent. In that sense, an ontology can be used to represent knowledge that is exchanged between the individual components of an agent. However, compared to other approaches for semantic representation, ontology-based data modeling usually shows some limitations, e. g., limited means for representing logical expressions like the quantifiers and their scope. Consider, for example, the following formula which contains a quantifier:

$$\forall x((P(x) \rightarrow Q(x) \rightarrow S(s)))$$

Even though there are several ways to express this by means of ontological instances, the corresponding ontological instances are very complex and more difficult to handle. The dialogue systems into which FADE has been integrated all make use of ontology-based meaning representation. In what follows, we will discuss three aspects of ontology-based representation of meaning and knowledge in multimodal dialogue systems: In the following section we will discuss representation languages and formats. Section 5.3 discusses how meta information can be represented and section 5.4 provides two practical examples of ontologies that have been employed in multimodal dialogue systems.

5.2 Representation Notations, Languages and Formats

There are a great number of representation languages but here we will focus only on a small subset, i. e., those that have an impact on the representation format that we use for FADE (see section 7.6). In what follows, we will discuss XML-based representation formats that are tailored for the representation of ontological data. At the end of this section we will discuss typed feature structures (Carpenter, 1992) and related operations in more detail, as this representation format provides the foundation of our implementation.

```

<rdf:RDF
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:sportevent="http://smartweb.semanticweb.org/ontology/sportevent#...">
<rdf:Description rdf:about="http://smartweb.semanticweb.org/ontology/sportevent#...">
  <rdf:type
    rdf:resource="http://smartweb.semanticweb.org/ontology/sportevent#FieldMatchFootballPlayer"/>
  <sportevent:inMatchTeam rdf:resource="http://smartweb.semanticweb.org/ontology/sportevent#..."/>
  <sportevent:number rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">11</sportevent:number>
  <sportevent:hasUpperRole rdf:resource="http://smartweb.semanticweb.org/ontology/sportevent#..."/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">[11] Doni, Cristiano</rdfs:label>
</rdf:Description>
</rdf:RDF>

```

Figure 5.1: Example of an RDF document describing an instance of a football-player in a specific match (the ellipses mark omitted information).

5.2.1 RDF

RDF is a simple metadata model that can be used to express statements about resources in the form of subject-predicate-object expressions. These expressions are called *triple*. This method for describing resources is an important aspect of the recent W3C activities concerning the promotion of the *semantic web*.

A key aspect of RDF is the resource identification. Each resource is named by a *Uniform Resource Identifier (URI)* which makes a resource directly identifiable (see figure 5.1). The URIs that name a resource do not have to be dereferenceable at all. From RDF, several ontology languages like RDFS and OWL (see below) have been developed, i. e., RDF is used to denote the instances of these ontologies.

5.2.2 RDF Schema

RDF Schema (RDFS) is an extensible framework for representing knowledge. It is defined by a W3C recommendation released in February 2004 (see <http://www.w3.org/TR/rdf-schema/>). RDFS is intended to structure RDF resources, and in that sense defines the type-hierarchy of an ontology.

The main constructs of RDFS are *classes* and *properties*:

- **rdfs:class** declares a resource to be a class. Hierarchical relations between classes can be defined by means of the **rdfs:subClassOf** element (see figure 5.2 for an example).
- **rdfs:property** defines a property (or slot) that can be attached to a set of classes that is defined via the **rdfs:domain** tag (or in other words: **rdfs:domain** declares the class of the *subject* of a triple using the property as predicate). The **rdfs:range** declares the class of the valid slot fillers of a predicate (i. e., it declares the class of the *object* of a triple).

5.2.3 Web Ontology Language—OWL

The Web Ontology Language (OWL) is a markup language designed to enable the exchange of data using an ontology. OWL is an extension of the RDF framework and is derived

```

<rdf:Description rdf:about=".../sportevent#FieldMatchFootballPlayer">
  <rdf:type rdf:resource=".../linginfo#ClassWithLingInfo"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    FieldMatchFootballPlayer</rdfs:label>
  <rdfs:subClassOf
    rdf:resource=".../sportevent#PositionalMatchFootballPlayer"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    A football player who is in the line-up of a team playing in a
    match.</rdfs:comment>
</rdf:Description>
<rdf:Description rdf:about=".../sportevent#number">
  <rdf:type
    rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:range
    rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
  <a:maxCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">
    1</a:maxCardinality>
  <a:range rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    integer</a:range>
  <rdfs:domain rdf:resource=".../sportevent#FieldMatchFootballPlayer"/>
  <rdfs:domain
    rdf:resource=".../sportevent#PositionalMatchFootballPlayer"/>
</rdf:Description>

```

Figure 5.2: Example RDFS excerpt defining the class *FieldMatchFootballPlayer* and one of its properties.

from the DAML+OIL ontology language. The specification of OWL is available at <http://www.w3.org/TR/owl-ref/> and there are a number of sublanguages like OWL Lite, OWL DL, and OWL Full. These three sublanguages show different levels of expressiveness and are designed for the use of specific groups of users and implementers.

OWL Lite supports those users who primarily need a classification hierarchy and simple constraints. For example, while it supports cardinality constraints, it only permits cardinality values of 0 or 1.

OWL DL supports those users who want maximum expressiveness while retaining computational completeness (all conclusions are guaranteed to be computable) and decidability (all computations will finish within a finite time). OWL DL includes all OWL language constructs but they can only be used under certain restrictions (for example, while a class may be a subclass of many classes, a class cannot be an instance of another class). OWL DL is so named because of its correspondence to description logics (see Horrocks (1998)), a field of research that studies the logics that form the formal foundation of OWL.

OWL Full is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. It is unlikely that any reasoning software will be able to support complete reasoning for every feature of OWL Full.¹

¹This enumeration is adopted from <http://www.w3.org/TR/owl-ref/> with minor changes.

5.2.4 Typed Feature Structures

Typed feature structures (TFS; see Carpenter (1992)) are an elegant way to represent complexly structured data so that it is still accessible to humans. TFSs are employed in various systems that deal with complex data. Moreover, as we will discuss later, our extended implementation of TFSs can be used to represent nearly all types of instances of ontologies (i. e., RDF).

Typed feature structures are based on the concept of feature structures and can be viewed as directed acyclic graphs (DAGs) where the nodes constitute objects and the paths, slots. Each object that follows a path is also a slot-filler. A feature structure-based representation of a song is depicted in figure 5.3.

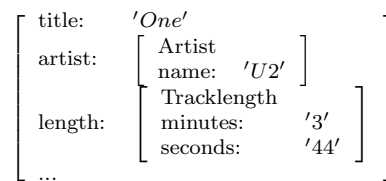


Figure 5.3: Feature structure based representation of a song.

For typed feature structures, each feature structure object is also assigned a type that defines its slot names and allowed slot-fillers (as in ontologies). Moreover, there is a type-hierarchy where a sub-type inherits all slot-definitions from its super-types. A typed feature structure-based representation of the song depicted in figure 5.3 is depicted in figure 5.4.

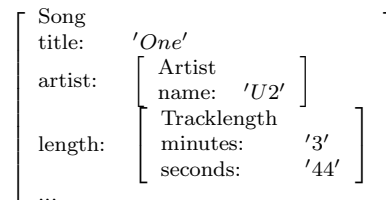


Figure 5.4: Typed feature structure based representation of a song.

Krieger mentions several benefits from using types, among them the ability to “abbreviate complex descriptions” and to “structure linguistic knowledge” as well as the constraining type checking property and the ability to use recursive type definitions (Krieger, 1995, p. 16). As we will discuss below, this is the fundament for operations like unification and overlay.

The following formal definition of TFSs is based on the one given in Alexandersson et al. (2006). This definition makes use of a notion of a type hierarchy where each pair of types has a *least upper bound* (LUB), i. e., a type that is the super-type of both types. In contrast, not all pairs of types have a *greatest lower bound* (GLB), which is defined as the most specific type that subsumes both types.

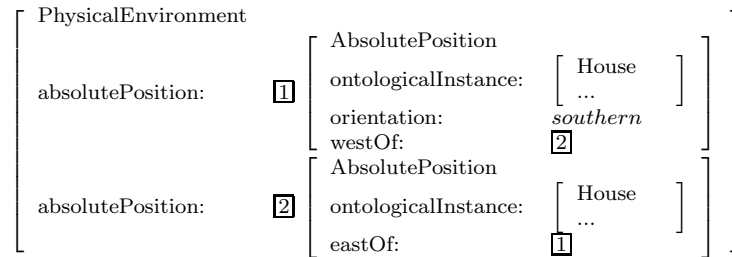


Figure 5.5: TFS exemplifying the co-reference mechanism of typed feature structures.

Definition 1 Partial Ordering of Types

A relation \sqsubseteq on a set of types - Type - is a partial order iff it is:

reflexive, i. e., $\forall x \in \text{Type}$ we have $x \sqsubseteq x$

antisymmetric, i. e., $\forall x, y \in \text{Type}$ if $x \sqsubseteq y \wedge y \sqsubseteq x$ then $x = y$

transitive, i. e., $\forall x, y, z \in \text{Type}$ if $x \sqsubseteq y \wedge y \sqsubseteq z$ then $x \sqsubseteq z$

Based on this partial ordering of Types $\langle \text{Type}, \sqsubseteq \rangle$ and a finite set of features Feat^2 :

Definition 2 Typed Feature Structures

A typed feature structure (TFS) is a tuple $F = \langle Q, \bar{q}, \theta, \delta \rangle$ where:

Q is a finite set of nodes with the root \bar{q}

\bar{q} is the (unique) root node

$\theta : Q \rightarrow \text{Type}$ is a total node typing function

$\delta : \text{Feat} \times Q \rightarrow Q$ is a partial feature value function

Carpenter (1992) introduced the so-called *appropriateness condition* which determines whether a feature is appropriate for a type and all its subtypes. Based on this notion of typed feature structures, there are two powerful operations that are used to compare and integrate two TFSs: *unification* and *overlay*. But before we can discuss these operations, we need to briefly consider the co-reference mechanism of TFSs.

Co-reference

TFSs also allow for a labeling of non-atomic objects. This makes it possible that two TFSs (possibly at different levels of nesting) can share their value. This kind of structure-sharing is called reentrancy or co-reference. Because TFSs in general allow reentrancy, they are not labeled trees, but Directed Acyclic Graphs (DAGs).

²Note that this definition of TFSs is restricted to bounded complete partial orders.

Unification

Unification is a commutative operation that takes two typed feature structures and returns a typed feature structure that covers the information of both TFSs if they are compatible and otherwise null. This means that $unify(a, b)$ where a and b are two typed feature structures will return the greatest lower bound of its two arguments (which is basically the most specific TFS that covers both arguments). Formally we define unification as follows:

Definition 3 Unification

In an inheritance hierarchy as described above, the unification of two feature structures is their greatest lower bound (GLB) \sqcup .

$$unify\left(\left[\begin{array}{l} \text{Song} \\ \text{title:} \\ \text{length:} \\ \dots \end{array} \left[\begin{array}{l} 'One' \\ \text{Tracklength} \\ \text{minutes:} \\ \text{seconds:} \end{array} \left[\begin{array}{l} '3' \\ '44' \end{array} \right] \right] \right], \left[\begin{array}{l} \text{Song} \\ \text{title:} \\ \text{artist:} \\ \dots \end{array} \left[\begin{array}{l} 'One' \\ \text{Artist} \\ \text{name:} \end{array} \left[\begin{array}{l} 'U2' \end{array} \right] \right] \right] \right) = \\ \left[\begin{array}{l} \text{Song} \\ \text{title:} \\ \text{length:} \\ \text{artist:} \\ \dots \end{array} \left[\begin{array}{l} 'One' \\ \text{Tracklength} \\ \text{minutes:} \\ \text{seconds:} \\ \text{Artist} \\ \text{name:} \end{array} \left[\begin{array}{l} '3' \\ '44' \\ 'U2' \end{array} \right] \right] \right]$$

Figure 5.6: Example of the unification of two TFSs.

Consider, for example, the unification of two compatible typed feature structures in figure 5.6. Here, the TFS on the right side of the equation comprises the information of both arguments of the unification operation.

But unification does not always succeed. If, on one hand, the two types of its arguments are not compatible, i. e., they do not subsume each other, unification will fail and return null (a condition we call *type clash*). If, on the other hand, two atomic values do not match, unification will also fail (a condition we call *conflicting values*). Consider, for example, the unification of the incompatible TFSs in figure 5.7.

$$unify\left(\left[\begin{array}{l} \text{Song} \\ \text{title:} \\ \text{length:} \\ \dots \end{array} \left[\begin{array}{l} 'One' \\ \text{Tracklength} \\ \text{minutes:} \\ \text{seconds:} \end{array} \left[\begin{array}{l} '3' \\ '44' \end{array} \right] \right] \right], \left[\begin{array}{l} \text{Song} \\ \text{title:} \\ \text{artist:} \\ \dots \end{array} \left[\begin{array}{l} 'Pride' \\ \text{Artist} \\ \text{name:} \end{array} \left[\begin{array}{l} 'U2' \end{array} \right] \right] \right] \right) = fail$$

Figure 5.7: Example of a failed unification.

Restricted Unification

A special case of unification is what we call *restricted unification* which is non-commutative in contrast to standard unification (i. e., $rUnify(a, b) \neq rUnify(b, a)$). The idea behind this restricted unification is that the first argument specifies the information (i. e., the slots) that at least must be comprised by the second argument. This operation is used for pattern matching purposes as we will see later. If we reconsider the last example—this time with matching atomic values, the restricted unification still fails, since the second argument does not contain any information about the track length (see figure 5.8).

$$runify\left(\left[\begin{array}{l} \text{Song} \\ \text{title:} \\ \text{length:} \\ \dots \end{array} \left[\begin{array}{l} 'One' \\ \text{Tracklength} \\ \text{minutes:} \\ \text{seconds:} \\ \dots \end{array} \left[\begin{array}{l} '3' \\ '44' \end{array} \right] \right] \right], \left[\begin{array}{l} \text{Song} \\ \text{title:} \\ \text{artist:} \\ \dots \end{array} \left[\begin{array}{l} 'One' \\ \text{Artist} \\ \text{name:} \\ \dots \end{array} \left[\begin{array}{l} 'U2' \end{array} \right] \right] \right] \right) = fail$$

Figure 5.8: Example of a failed restricted unification.

In Chapter 7 we will see how this operation can be used for pattern matching in a production rule system.

Overlay

Overlay is a non-commutative operation on typed feature structures introduced by (Alexandersson and Becker, 2001, 2003; Alexandersson et al., 2006). It is based on unification, involving two TFS called *covering* (co) and *background* (bg) to create a TFS containing the information from the covering and equipping it with additional information from the background. As opposed to unification, overlay never fails and at least returns the covering. Where unification would have failed because of a type-clash, overlay results in a possible loss of background information (and also lower scoring).

Following Alexandersson and Becker (2001), the overlay operation consists of two basic steps:

Assimilation: Two TFSs can only be integrated if the two are in a direct subsumption relation. In case of a type-clash the background needs to be refined to the most specific supertype of the two TFSs, a process that is called *assimilation*. Note that the type of the two involved TFSs will not be affected by this operation.

Overlay: The actual process of overlaying two TFSs is very similar to the unification operation, however, with the difference that in case of conflicts the information from the covering overwrites the background.

Since overlay always returns a result, it is necessary to assess the outcome, i. e., to provide a score for the resulting TFS that reflects how well the two TFSs fit together. Pfleger et al. (2002) present a function that computes the score of an overlay quality which will be briefly recapitulated at the end of this section. To understand how overlay functions, it might be useful to think of putting two shapes on top of each other. If we now view the resulting

shape from the top, we can see all information stemming from the shape on the top plus all information that is not covered by the shape on top, where the (conflicting) yellow information from the background is covered by the information of the covering. Technically, overlay functions just like unification. However, if a type-clash or conflicting values occur, it operates as follows:

- **Type-clash:** In this case, overlay computes the least upper bound (LUB) of the two involved types (a process that is called assimilation, i.e., the type of the background is assimilated to the most specific common super-type of the two types). Then, the overlay of the covering and the assimilated background is computed.
- **Conflicting values:** If both the covering and the background contain conflicting information in an atomic object (e.g., different song titles), the value of the covering overwrites the value of the background.

An example where unification would fail but overlay succeeds is given in figure 5.9. Here the conflicting atomic value of the background (“Pride”) is overwritten with the information of the covering for that feature (“One”).

$$\text{overlay}\left(\left[\begin{array}{l} \text{Song} \\ \text{title:} \\ \text{length:} \\ \dots \end{array} \left[\begin{array}{l} 'One' \\ \text{Tracklength} \\ \text{minutes:} \\ \text{seconds:} \end{array} \left[\begin{array}{l} \\ '3' \\ '44' \end{array} \right] \right] \right], \left[\begin{array}{l} \text{Song} \\ \text{title:} \\ \text{artist:} \\ \dots \end{array} \left[\begin{array}{l} 'Pride' \\ \text{Artist} \\ \text{name:} \\ 'U2' \end{array} \right] \right] \right) =$$

$$\left[\begin{array}{l} \text{Song} \\ \text{title:} \\ \text{length:} \\ \text{artist:} \\ \dots \end{array} \left[\begin{array}{l} 'One' \\ \text{Tracklength} \\ \text{minutes:} \\ \text{seconds:} \\ \text{Artist} \\ \text{name:} \\ 'U2' \end{array} \right] \right]$$

Figure 5.9: Example of overlay where unification would fail.

Scoring of Overlay Results A key feature that separates overlay from classic default unification is the built-in scoring mechanism. Every time overlay is applied to two TFSs, the algorithm also computes a score that reflects the quality of the result. This is particularly interesting, as overlay never fails and would thus provide a result even if no background information could be added to the covering. At the moment, there are two different kinds of scoring mechanisms for overlay: (i) a scoring function that reflects the structural consistency of the involved arguments (see Pflieger et al. (2002)), and (ii) a scoring function that reflects the informational distance between the two arguments (see Alexandersson et al. (2004)).

Structural Consistency In general, the overall score of an overlay operation should reflect how well the covering fits the background in terms of non-conflicting features. Another important point which should be covered by the scoring mechanism is the occurrence of a

type clash between two features. In this case, a unification of two feature structures would fail and this has to be expressed by a lower score. These assumptions lead to a heuristic that is based on the contrast between the amount of non-conflicting features and the amount of conflicting features (including type-clashes). This heuristic uses four scoring parameters (initialized to zero). During overlay these parameters are incremented as indicated below:

co a feature or a (atomic) value in the result stems from the covering. **co** is incremented for each feature stemming from the covering.

bg a feature or a (atomic) value that also occurs in the background. **bg** is incremented for each feature stemming from the background.

tc type clash, i.e., the type of the covering and background was not identical. This is identified during the computation of the assimilation.

cv conflicting values. This occurs when the value of a feature from the background is overwritten.

The sum of **co** and **bg** minus the sum of **tc** and **cv** will be weighted by the sum of **co**, **bg**, **tc** and **cv**. This leads to a function shown in the following formula whose co-domain is $[-1,1]$:

Definition 4 (Score)

$$score(co, bg, tc, cv) = \frac{co + bg - (tc + cv)}{co + bg + (tc + cv)}$$

The positive extreme ($score(co, gb, tc, cv)=1$) indicates that the feature structures are unifiable. The negative extreme ($score(co, gb, tc, cv)=-1$) indicates that all information from the background has been overwritten by information from the cover. A score within this interval indicates that the cover more or less fits the background: the higher the score is, the better the cover fits the background. Negative values signal that conflicting and thus overlaid values outweigh unifiable values (positive values vice versa). Applied to the example depicted in figure 5.9, the scoring parameter of overlay would be:

$co = 4$; four features stemming from the covering (the features: title, length, minutes, sec).

$bg = 3$; three features stemming from the covering (the features: artist, firstName, lastName).

$tc = 0$; no type clash occurred.

$cv = 1$; one conflicting value for the feature "title."

Applied to the scoring function this results in:

$$score(co, bg, tc, cv) = \frac{4 + 3 - (0 + 1)}{4 + 3 + (0 + 1)} = \frac{6}{8} = 0.75$$

Note that only *filled* slots are considered features when collecting the scoring parameters co and bg . Any empty feature that has no value is not considered. The example in figure 5.10 illustrates this and also shows what happens in case of conflicting types. This simplified example assumes a type hierarchy for entertainment where the type *Movie* and the type *News* have a common super-type *Broadcast* and share a set of features (e. g., *title*, *channel*, *startTime*, *endTime*). The type *Movie*, for example, also comprises a feature *cast* that is not available for the type *News*.

$$\text{overlay}\left(\left[\begin{array}{l} \text{News} \\ \\ \\ \end{array}\right], \left[\begin{array}{l} \text{Movie} \\ \text{cast:} \left[\begin{array}{l} \text{Person} \\ \text{firstName: } 'Arnold' \\ \text{secondName: } 'Schwarzenegger' \end{array}\right] \\ \text{startTime:} \left[\begin{array}{l} \text{TimePoint} \\ \text{hour: } '20' \\ \text{minute: } '00' \end{array}\right] \end{array}\right]\right) =$$

$$\left[\begin{array}{l} \text{News} \\ \text{startTime:} \left[\begin{array}{l} \text{TimePoint} \\ \text{hour: } '20' \\ \text{minute: } '00' \end{array}\right] \\ \text{endTime: } \dots \end{array}\right]$$

Figure 5.10: Overlay of two TFSs that have conflicting types.

The application of *overlay* in figure 5.10 directly leads to a type clash which normal unification would not be able to deal with. But *overlay* is able to resolve this conflict and returns an enriched representation of the covering. The scoring for the result is based on the following parameters:

$co = 0$; no feature stemming from the covering (the empty features do not count).

$bg = 1$; one feature stemming from the covering (the feature *startTime*).

$tc = 1$; one type clash occurred.

$cv = 0$; no conflicting values.

Applied to the scoring function this results in:

$$\text{score}(co, bg, tc, cv) = \frac{0 + 1 - (1 + 0)}{0 + 1 + (1 + 0)} = \frac{0}{2} = 0$$

Note that this example is kept very simple. Usually one would choose a more complex representation for the values of *startTime* (e.g. a complex type *TimeExpression* that again consists of several features).

Informational Distance Besides structural consistency, the scoring of an *overlay* result can also be used to express the several other relations that hold between the two arguments of *overlay*. For example, the original scoring function does not reflect the severeness of a type clash, i. e., the geometric distance between the two types of the involved objects.

In Alexandersson et al. (2004) we defined a second scoring function for overlay called the *informational score*. This scoring function is based on the notion of *informational distance*, which reflects the ratio of lost and kept features of the background:

Definition 5 (Informational Distance)

Let

bg be the type of the background of the type clash,

lub be the type of the least upper bound of the clashed cover and background,

$|lub|$ be the number of features defined for the LUB type,

$|bg|$ be the number of features defined for the background type.

Then, the *informational distance*— $idist$ —is defined as:

$$idist(lub, bg) = \begin{cases} 0 & \text{if } |bg| = 0 \\ \frac{|bg| - |lub|}{|bg|} & \text{otherwise} \end{cases}$$

Using this informational distance, we are able to revise our original scoring function with *weighted type clash* that replaces the original counter for type clashes:

Definition 6 (Weighted Type Clash)

Let

co be number of TFSs or atomic features stemming from the covering,

bg be the number of feature values or atomic features stemming from the background,

i, \dots, n be type clashes,

cv be the number of conflicting atomic values.

Then, the *weighted type clash* (wtc) is defined as

$$wtc = \sum_{i=1}^n idist(lub_i, bg_i)$$

Thus, the extended scoring function looks like the following:

Definition 7 (Informational Score)

$$score(co, wtc, bg, cv) = \frac{co + bg - (wtc + cv)}{co + bg + wtc + cv}$$

5.2.5 Discussion

Typed feature structures provide a formal framework for dealing with complex nested data structures. As we have discussed in Alexandersson and Pflieger (2006); Alexandersson et al. (2006), TFSs and operations like unification and overlay provide an elegant way of dealing with context-dependent information. However, standard TFSs do not support unique resource identifiers that are commonly used in ontologies to permit the identification of unique entities. As we will show in section 7.6.2, we have extended the concept of TFSs with unique identifiers so that it is possible to mark and identify unique real world objects.

Using the overlay operation, it is possible to integrate partially incompatible information. However, the application of overlay (or default unification in general) shows several drawbacks when dealing with instances of real world objects. Consider, for example, the application of overlay to a TFS describing a partial address and a TFS describing a complete address³:

```

<object type="Address">
  <slot name="GEOPOSITION">
    <object type="WGS84GeoPosition">
      <slot name="LATITUDE">
        <value type="String">52.5167</value>
      </slot>
      [...]
    </object>
  </slot>
  <slot name="POSTALCODE">
    <value type="String">13353</value>
  </slot>
  <slot name="ROADNAME">
    <value type="String">M"ullerstrasse 163 b</value>
  </slot>
  <slot name="HOUSENUMBER">
    <value type="String">1 - 9999</value>
  </slot>
  <slot name="hasCity">
    <object type="City">
      <slot name="HAS-DENOMINATION">
        <object type="denomination">
          <slot name="NAME">
            <value type="String">Karlsruhe </value>
          </slot>
        </object>
      </slot>
    </object>
  </slot>
</object>

```

```

<object type="Address">
  <slot name="GEOPOSITION">
    <object type="WGS84GeoPosition">
      <slot name="LATITUDE">
        <value type="String">52.5167</value>
      </slot>
      [...]
    </object>
  </slot>
  <slot name="POSTALCODE">
    <value type="String">13353</value>
  </slot>
  <slot name="ROADNAME">
    <value type="String">M"ullerstrasse 163 b</value>
  </slot>
  <slot name="HOUSENUMBER">
    <value type="String">1 - 9999</value>
  </slot>
  <slot name="hasCity">
    <object type="City">
      <slot name="HAS-DENOMINATION">
        <object type="denomination">
          <slot name="NAME">
            <value type="String">Berlin</value>
          </slot>
        </object>
      </slot>
    </object>
  </slot>
</object>

```

Overlaying the left TFS with the one on the right would result in a semantically incompatible TFS:

```

<object type="Address">
  <slot name="GEOPOSITION">
    <object type="WGS84GeoPosition">
      <slot name="LATITUDE">
        <value type="String">52.5167</value>
      </slot>
      [...]
    </object>
  </slot>
  <slot name="POSTALCODE">

```

³These example structures are taken from the SMARTWEB ontology SWINTO (see section 5.4.2).

```

    <value type="String">13353</value>
  </slot>
  <slot name="ROADNAME">
    <value type="String">M"ullerstrasse 163 b</value>
  </slot>
  <slot name="HOUSENUMBER">
    <value type="String">1 - 9999</value>
  </slot>
  <slot name="hasCity">
    <object type="City">
      <slot name="HAS-DENOMINATION">
        <object type="denomination">
          <slot name="NAME">
            <value type="String">Karlsruhe</value>
          </slot>
        </object>
      </slot>
    </object>
  </slot>
</object>
</slot>
</object>

```

Generally, this type of semantically inconsistent TFSs can occur with all kinds of real world objects and this has to be taken into account when overlay is employed. However, at this point, the boundaries between representation languages and semantics blur. From the perspective of the representation format, such a TFS is valid.

Another important aspect that needs to be considered when using overlay is the processing of multiple occurrences of the same slot as in sets. In contrast to the normal processing where each slot pair is overlaid recursively, sets or lists require a different processing strategy. E. g., for some kinds of sets it is more suitable to compare the individual elements of the sets while other kinds should be dealt with by standard set operations like union. Romanelli et al. discuss this challenge from a discourse processing perspective and identify five different types of set operations that are triggered by lexical markers (see Romanelli et al. (2005)).

5.3 Modeling Meta Information

Another crucial issue in multimodal dialogue systems is the representation of intermediate processing results as they are generated by the individual components. To this end, a number of standardized communication protocols have been put forward that allow the representation of different aspects of meta-information in addition to the task-specific meaning of an utterance or event. The main purpose of these communication protocols is to provide means to annotate the actual content of user input with timestamps, confidence values, etc. Moreover, these protocols are also suited to represent multiple interpretation hypotheses in case of ambiguous user utterances. In the following, we will briefly introduce two of them, namely M3L and EMMA:

M3L (see Herzog and Ndiaye (2006)): The *Multimodal Markup Language* (M3L) serves as the external data format for an information exchange between the functional components of the SMARTKOM project (Wahlster, 2006). It is an XML-based representation language and employs XSLT stylesheets for the transformation of the internal formats of the components and the external communication format. M3L is defined by a set of XML schemas⁴ that are automatically generated from an underlying ontological taxonomy. This taxonomy is represented in the OIL format (Fensel et al., 2001). Gurevych

⁴See <http://www.w3.org/XML/Schema/>.

et al. (2003a) introduce an offline tool that automatically incorporates an ontology represented in OIL into the M3L specification and generates the corresponding XSLT schemas.

M3L provides means to represent monomodal as well as multimodal user input but also presentation tasks or commands for external applications. The following simplified excerpt of an M3L document shows an intention lattice representation of the multimodal user input “*I would like to know more about this*” [pointing gesture at a movie displayed on the screen]:

```
<intentionLattice>
  [...]
  <hypothesisSequence>
    <hypothesisSequence>
      <inputText>
        I would like to know more about this.
      </inputText>
      <score>
        <source> acoustic </source>
        <value> 0.96448 </value>
      </score>
      <hypothesis>
        <discourseStatus>
          <discourseAction> set </discourseAction>
          <discourseTopic> <goal> epg_info </goal> </discourseTopic>
        </discourseStatus>
        [...]
        <event id='dim868''>
          <pieceOfInformation>
            <broadcast id='dim863''>
              <avMedium>
                <avMedium id='dim866''>
              [...]
            </pieceOfInformation>
          </event>
        </hypothesis>
      </hypothesisSequence>
    </hypothesisSequence>
  </intentionLattice>
```

Using the `<hypothesisSequence>` tags, it is possible to represent multiple possible interpretations for the same input event. Each individual interpretation hypothesis can be annotated with confidence values using the `<score>` tag.

EMMA (see <http://www.w3.org/TR/emma/>): The *Extensible MultiModal Annotation* language (EMMA) is intended to be used as an annotation language for systems that provide semantic interpretations for speech, typed text, and ink input. It is being developed by the W3C Multimodal Interaction Working Group and had the status of a Last Call Working Draft at the time of writing. EMMA can be used to specify the medium, function and mode of an input event, as well as various timestamps (e.g, start, end, duration) and confidence values. Consider, for example, the following EMMA document which represents an ambiguous speech recognition result:

```
<emma:emma version="1.0"
  xmlns:emma:="http://www.w3.org/2003/04/emma"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/2003/04/emma
  http://www.w3.org/TR/emma/emma10.xsd"
  xmlns="http://www.example.com/example">
  <emma:one-of id="r1" emma:start="1087995961542" emma:end="1087995963542">
```

```

<emma:interpretation id="int1" emma:confidence="0.75"
  emma:tokens="flights from boston to denver">
  <origin>Boston</origin>
  <destination>Denver</destination>
</emma:interpretation>
<emma:interpretation id="int2" emma:confidence="0.68"
  emma:tokens="flights from austin to denver">
  <origin>Austin</origin>
  <destination>Denver</destination>
</emma:interpretation>
</emma:one-of>
</emma:emma>

```

The two interpretation hypotheses are embedded under the tag `<emma:one-of>`. The attribute `emma:confidence` specifies for both interpretation hypotheses the corresponding confidence value of the recognizer and the actual semantic representation of the interpretation is embedded under the `emma:interpretation` tag. In addition to representing N-best lists of interpretations, EMMA can also be used to represent lattices of words or symbols using the `emma:lattice` tag.

EMMA has been designed for primarily representing user input. However, EMMA has been extended within the SMARTWEB project in order to fit the purpose of representing system results as well. The resulting markup language is called SWEMMA (Reithinger et al., 2005a; Sonntag and Romanelli, 2006).

5.4 Ontology-based Representation of Meaning

As Gurevych et al. (2006) highlight, there are many dialogue systems that employ heterogeneous knowledge representation formats for their individual components (Gurevych et al., 2006, p. 71). This means that such systems have no system-wide notion of how to represent, for example, a city. Gurevych et al. argue for a homogeneous world model that serves as a common representation format for all components of the system and that both represents general conceptualizations (via a top-level or generic ontology) and particular domains (domain-specific ontologies).

There are many ways of defining an ontology for a multimodal dialogue system and in practice, the contents of an ontology are often driven by all kinds of system requirements. However, the general idea to use the ontological concepts as semantic representations of the participants' utterances remains stable. In what follows, we will very briefly discuss two ontologies that have had an impact on the present work. It should be noted though, that this section reflects only a small and very subjective excerpt of the work concerning ontologies.

5.4.1 The SmartKom Ontology

The SMARTKOM ontology (Gurevych et al., 2006) was developed for the SMARTKOM project (see Wahlster (2006)). The top-level ontology of SMARTKOM follows the distinction between entities with *primary ontological status* and roles taken by them in specific situations (see Guarino and Welty (2000)). An example would be the class *Person* which is a primary entity. A *FootballPlayer*, however, is a *Person* that takes the particular role of a football player.

This distinction between primary ontological entities and the roles they can take is realized at the top-level of this ontology by means of the two classes *Type* and *Role*. The other classes of SMARTKOM's upper-level model are depicted in figure 5.11.

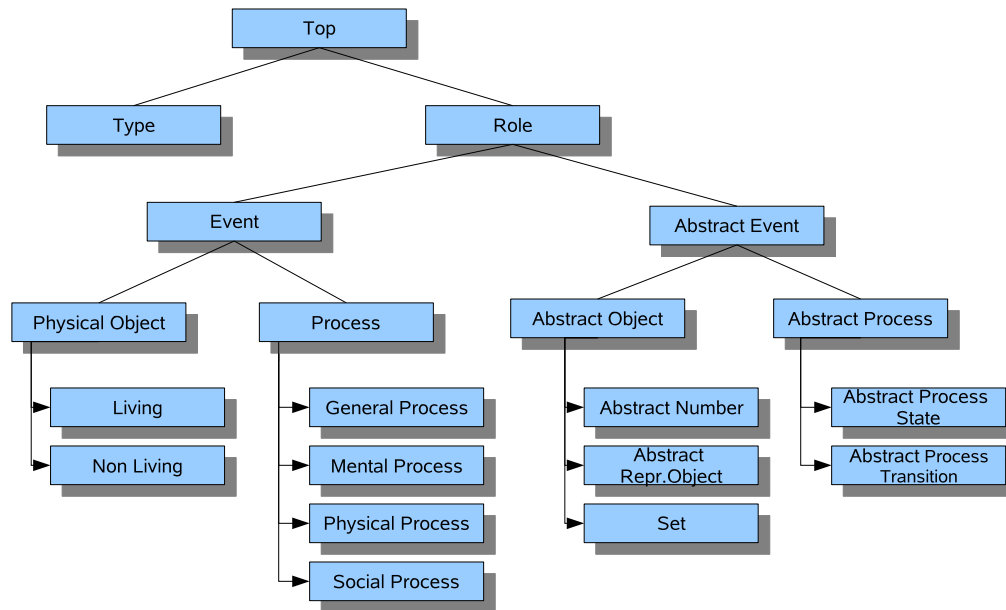


Figure 5.11: Top-level part of the SMARTKOM ontology.

The *Role* class has two direct subclasses: *Event* and *AbstractEvent*. All subclasses of *Event* describe roles of entities or processes that occur in the real world while all subclasses of *AbstractEvent* describe abstract categorizations and descriptions (e. g., *Sets*, *SpatialRelation*). Events in turn are further classified into *PhysicalObject* and *Process*. Domain knowledge in SMARTKOM is described by means of a frame-based hierarchical representation of real world entities. Consider, for example, the following XML-based representation of a cinema:

```

<cinema id="mf1330">
  <movieTheater id="mf1331">
    <entityKey id="mf1338"> movieTheater_1000006 </entityKey>
    <name id="mf1315"> Studio Europa </name>
    <contact id="mf1332">
      <address id="mf1333">
        <town id="mf1334"> Heidelberg </town>
        <geoCoordinate id="mf1335">
          <x id="mf1336"> 3477692.0 </x>
          <y id="mf1337"> 5473931.0 </y>
        </geoCoordinate>
      </address>
    </contact>
  </movieTheater>
</cinema>

```

On the top-level, this specific instance of a cinema is described by means of a *entityKey*, a *name* and a *contact* address. The *entityKey* slot contains the index of the database for that object and indicates that this instance is a fully specified instance. However, in multimodal dialogue systems most of the instances that describe the analysis results for user utterances consist of underspecified instances. Consider for example a user request like “*What movies are on at the cinema Studio Europa?*” After the analysis of this utterance has been finished, the ontological object describing the cinema looks as follows:

```

<cinema id="mf2230">
  <movieTheater id="mf2231">
    <name id="mf2215"> Studio Europa </name>
  </movieTheater>
</cinema>

```

This ontological object is underspecified since it is not clear which instance is actually referred to. If there is more than one cinema of that name and there is no contextual information that would help to disambiguate the request, it is not possible for the system to determine which cinema was actually meant by the user.

5.4.2 The SmartWeb Ontology

All knowledge in the SMARTWEB system is represented by means of a single system-wide ontology called *SWINTO* (SMARTWEB INTeGrated Ontology; see Oberle et al. (2006)). This ontology integrates central concepts of SUMO (Niles and Pease, 2001) and DOLCE (Gangemi et al., 2002) and covers a set of sub-ontologies. These sub-ontologies model the domain-specific concepts (see also figure 5.12):

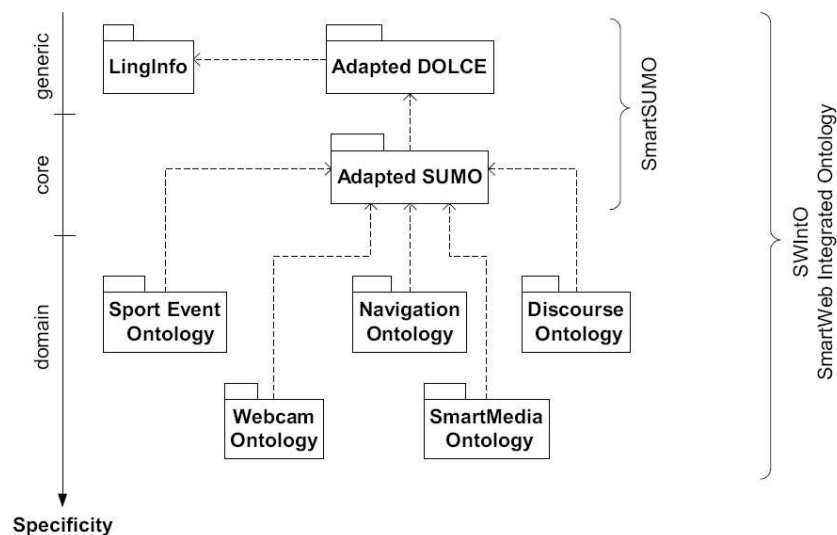


Figure 5.12: Overview of SmartSUMO/SWIntO as a UML package diagram. Packages represent ontology modules; dotted lines represent dependencies between modules (taken from Oberle et al. (2006)).

SportEvent Ontology—modeling concepts related to the domain of sports focusing on football and the World Cup. This ontology comprises about 50K instances modeling all World Cup events since 1954 (including players, games, results, etc.).

Navigation Ontology—modeling concepts required for navigation and routing tasks (e. g. points-of-interest, cities, maps).

Discourse Ontology—modeling dialogue-specific concepts that are used by the individual components of SMARTWEB’s multimodal dialogue system.

SmartMedia Ontology—modeling concepts for representing pictures, movies, and other multimodal information.

LingInfo Ontology—modeling a linguistic knowledge base providing linguistic information (orthographic realizations, grammatical gender, stem and inflection) for all ontological classes and properties of the SMARTWEB ontology.

A key feature of the SMARTWEB ontology is that the classes and properties can be annotated with linguistic information using the LingInfo meta-ontology. Typically, as Buitelaar et al. (2006) argue, “such information is mostly missing or represented in impoverished ways, leaving the semantic information in an ontology without a grounding to the human cognitive and linguistic domain” (Buitelaar et al., 2006, p. 28). Using the LingInfo ontology, it is possible to annotate individual classes with language-specific information. Figure 5.13 provides an example of the linguistic annotations for two subclasses of the class *FootballPlayer*. In this example, the class *Defender* is annotated with the term *Abwehrspieler* for German. In addition to that, it is also possible to annotate expressions with their linguistic features like gender, number, case. As we will discuss in section 8.2, we have incorporated parts of the LingInfo ontology into the representation of linguistic features of referring expressions.

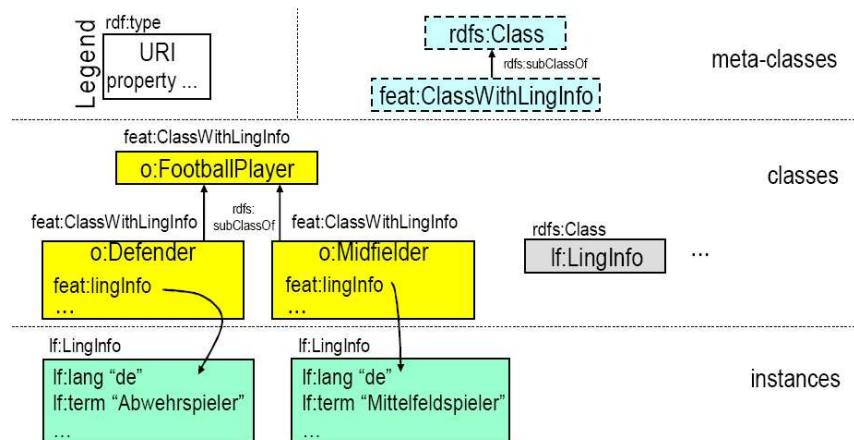


Figure 5.13: LingInfo model with example domain ontology classes and LingInfo instances (simplified) (taken from Buitelaar et al. (2006)).

5.5 Conclusion and Discussion

In this chapter we have discussed the central aspects of representing meaning and meta information for multimodal dialogue systems. We have introduced common representation formats like RDF and OWL and discussed Carpenter’s typed feature structures (TFSs) in

detail. Moreover, we have considered two central operations for manipulating TFSs: unification and overlay. As mentioned, these operations are of particular importance for dealing with complex ontological data. We also introduced two representation languages for modeling meta information that go beyond the actual meaning of contributions of human users and virtual agents. Finally, we provided two examples of practically employed ontologies.

The question of whether it is possible to represent ontological data by means of TFSs is very important for our work. In principal, as discussed in Alexandersson and Becker (2001); Alexandersson et al. (2006), typed feature structures can be used to represent ontological data. The mapping is rather straightforward, ontological classes can be mapped to types, and features to slots. Both representation frameworks support inheritance networks for their type-hierarchy. However, there are some concepts in the ontological representation frameworks that cannot be directly mapped to typed feature structures. First of all, the TFS framework does not model concrete and unique instances of a class. Consider as an example, an instance of the class *Person*: as long as the instance does not describe any specific person there is no problem. However, as soon as an ontological instance describes a specific person (e. g., the German football player Michael Ballack), this instance is assigned a unique resource identifier (URI) which can be used to reference this instance. In TFS there are means to realize such unique entities. OWL introduces two other aspects that cannot be represented with TFS. First, it is possible to state cardinality restrictions for slot-fillers with TFS and second, OWL allows a further specialization of slot-fillers in the inheritance network.

For the work presented here, we adopted the typed feature structure framework as a starting point and extended it with the concept of unique resources representing unique entities in the real world. Our view of ontological data is based on *extended typed feature structures* (see section 7.6.2) where TFS can have identifiers that are not only valid within a TFS document, but also across documents. What has only been briefly mentioned is the issue of how to deal with sets and lists in a TFS-based approach. In general, TFSs are not designed to cope with set-like structures, especially when we consider unification or overlay. Moreover, there is another issue still open with overlaying instances of real world objects, as this can result in inconsistent objects. We will outline a solution to this issue in section 7.6.2.

Chapter 6

A Comprehensive Context Model for Multimodal Multiparty Discourse

This chapter presents and discusses a comprehensive context model for multimodal-multiparty dialogues. This model is designed to support both the task of multimodal fusion and the task of discourse processing in a multimodal dialogue system. Section 6.1 gives an overview of the basic contextual categories of the model and discusses the difference between the immediate conversational context and the discourse context. Section 6.2 and section 6.3 then discuss the structure of the conversational and the discourse context respectively, in detail.

6.1 Introduction

Based on the notion of *local context factors* discussed in (Bunt, 2000) (see also section 4.2), we differentiate five categories of contextual factors that need to be represented in order to be able to deal with the full range of multimodal discourse phenomena mentioned in Chapter 3¹:

Physical Context—objects that are present in the surroundings and that might serve as potential referents

Perceptual Context—general events or actions conducted by the other participants; projected/expected actions of the participants

Conversational Context—current status of the conversation with respect to turn management; current conversational roles of the participants (speaker, addressee, overhearer, eavesdropper)

Linguistic Context—discourse history; including unique representations of referents

Social Context—social roles of the participants.

¹Note that even though the names of some categories resemble the ones used in (Bunt, 1994, 2000), our connotation is slightly different. For example, when Bunt refers to the physical context, he means the available communication partners and channels while we see the physical context as the actual physical objects present in the surroundings.

6.1.1 Taking the Perspective of the Participants

Typically, the context model of a dialogue system represents context from a bird-eye perspective, i. e., just like an impartial recorder who tracks the individual actions of the participants. The key to our approach is that the context model is tailored to the view that each individual participant has of the interaction. This means that the context model will always reflect an incomplete representation of what happened, or as Bunt highlighted: “There is no room here for an ‘objective’ notion of context, since the participants’ communicative behavior depends solely on how they view the situation, not on what the situation really is” (Bunt, 2000, p. 101). However, this subjective notion of context is only applicable for the discourse context. As we discuss below, there are aspects of context that provide an objective view (i. e., the representation of the physical surroundings).

6.1.2 Immediate Conversational Context vs. Discourse Context

The distinction between *interactional* and *propositional* information in contributions (see Cassell et al. (1999)) is of particular importance for our context model as these two types of contributions require different processing strategies. Interactional information contributes to the structural organization of the conversation as it regulates the exchange of turns, helps to avoid overlapping speech, is used to provide backchannel feedback and supports the identification of the intended addressees of a contribution.

Based on this distinction, we differentiate between two types of context representations that an artificial participant (i. e., a dialogue system or a virtual character) in a conversation needs to maintain. The first is the *immediate conversational context* representing the current physical and perceptual context. This immediate turn context serves as a temporal storage for perceived monomodal events that need to be interpreted in their context of use (this approach is derived from Pflieger (2004)). The second type is a long-term *discourse context* representing previous contributions of the participants. This discourse model supports the resolution of referring expressions by means of referents derived from accompanying gestures and those introduced in the previous discourse.

While interactional nonverbal behavior (such as head nods, gazing, beat gestures) is incorporated into the representation of the immediate turn context, pointing gestures and iconic gestures are incorporated into the discourse context. These latter gestures are typically resolved by a multimodal fusion component but in our approach they are processed together with the spoken referring expressions. Formally, we define the context Γ of interactions as pairs of $\langle IC_A, DC_A \rangle$, where IC_A is the immediate conversational context as it is perceived by a participant A , and DC_A is the discourse context as it is perceived by participant A .

6.2 The Immediate Conversational Context

The purpose of the immediate conversational context is to maintain an effective representation of the conversational status so that perceived monomodal events can be interpreted with respect to their impact on the interactional development of the conversation. The immediate conversational context comprises aspects of the physical context, perceptual context, conversational context and social context from the perspective of an individual participant of an interaction. The structure of the conversational context is centered around the physical context, the individual participants and their current actions. But this context represents more

information than the simple presence of physical objects and participants in the surrounding. For example, besides the physical properties, the representations of the participants comprises also their current conversational role, their current active interactional signals (e.g., gaze, gesticulation) and their social status (if available). Additionally, the participants have a rich representation of themselves including their individual aims.

This conversational context also builds the basis for any reactive behavior of the system. All perceived monomodal and so far uninterpreted events are categorized and integrated into this context model. This permits direct reactions to events that are of particular impact for that participant that a context representations stands for. If, for example, someone else starts to speak, this is immediately registered in the conversational context which in turn prevents our participant from unintentionally interrupting the current speaker.

The purpose of the immediate conversational context is to always reflect the current state of the conversation. This has the consequence that it has to be updated as frequently as possible. Moreover, the conversational context does not reflect the history of events but rather provides a snapshot of the current state of affairs. Any information that is needed later on has to be stored in the discourse model.

Formally, we define the immediate conversational context IC_A of a participant A as a set of $\langle CS, P_i, \dots, P_n \rangle$, where CS represents the conversational status and P_i a participant of the conversation. In the remainder of this section, we will describe in detail the model of the immediate conversational context. However, note that this information is to some extent tailored to the needs of the task at hand and is subject to extensions or reductions if other tasks require more or less details.

6.2.1 Representing Dialogue Participants

Each participant of an interaction typically has a good notion of the other participants present and uses this information in various ways both for processing and generating contributions. Part of this information can be directly perceived from the participants' appearance (e.g., the sex, sometimes the social status) while other information can only be projected or collected during the course of the interaction. But more importantly, only parts of this information is really relevant for conducting the interaction itself.

The fact, for example, that someone is left-handed or has black hair usually has no direct impact on the conversation. However, this leaves us with the question of what is relevant information about the participants of a dialogue? When we reconsider the aspects of multimodal multiparty interactions we discussed in Chapter 3, we get an answer to this question that fits our purposes.

When we start with the addressee identification, the name and sex of a participant support the identification of the current roles of the participants. But of course, gazing behavior also contributes a lot to determining the intended addressee of an utterance. In general, any nonverbal behavior of a participant should be represented in the conversational context, as these are considered strong turn-taking signals (see section 3.3.2). Regarding place deixis or spatial references, the participant's preferred or currently employed frame of reference is an important information as speakers tend to tailor their contributions to that of the addressees and use the same frame of reference. Additionally, the current location of a participant influences the resolution of spatial referring expressions and should therefore be represented in the conversational context.

Membership in communal groups (see section 3.2.5), expertise and the social status of a

participant also contribute to the processing and generation of dialogue contributions but in a less direct way than the aforementioned aspects. As discussed in section 3.2.5, this information is used to select appropriate and adapted referring expressions and phrases. Moreover, the perceived emotional state of a participant also has some impact on the interpretation of an utterance (i. e., irony, sarcasm).

To summarize, the representation of a participant in the immediate conversational context comprises the following aspects:

Name: The participant's first and last name (but also nicknames).

Sex: The participant's sex (*male*, *female*).

Nonverbal behavior: The currently active nonverbal behavior represents the list of currently perceived nonverbal behavior of that participant.

Frame of Reference: The currently used frame of reference intrinsic, relative or absolute (see section 3.2.3).

Position: The participant's position in the scene (i. e., the top-level physical environment).

Emotional state: The participant's emotional state (if available).

Communal groups: Assumed membership in communal groups.

Expertise: Assumed expertise of the participant.

Social status: Assumed social status of the participant.

If the participant to be represented is a human user, part of this information is represented in so-called *user models* (see for example Wahlster and Kobsa (1989); Heckmann (2006)). Usually, a user model encompasses information and assumptions about all aspects of the user that might affect the interaction. Many dialogue systems comprise an explicit *user modeling* component that provides this kind of information for other components. In that way, the representation of a human participant in the immediate conversational context can be provided by a user model.

6.2.2 Modeling the Conversational Status

While interacting with other people, humans usually have a clear understanding of the conversational status, i. e., they know who the speaker is, what their own role is, or if they are supposed to speak. As we discussed in Chapter 3, this understanding is crucial for a successful interaction between two or more interlocutors. To this end, our context model comprises a representation of the *conversational state* which can be filled with a wide variety of information. However, besides information that is motivated by the empirical work we discussed in Chapter 3, this information can also be of a rather technical nature. E. g., as we will discuss in section 9.3, the conversational state of the SMARTWEB system comprises rather artificial values like the probability that the user addressed the system by means of *OnTalk* and *OnView* measures.

As discussed in section 3.3.2, participating in multiparty interactions requires a participant to carefully monitor the turn-taking signals of the current speaker (e. g., pauses, rising or

falling pitch). Moreover, in order to be able to identify the addressee(s) of an utterance, a participant also needs information about the current and previous speakers and addressees. In order to support the corresponding reasoning processes, we envision the following information in the conversational status of our context model:

Characteristics of the current turn:

Turn duration—how long is the current speaker already holding the turn.

FrameOfReference—holds currently active frame of reference.

TemporalReferencePoint—currently activated temporal reference point; the last time point mentioned in the interaction.

Prosody—represents falling pitch at the end of a sentence or the drawl of a syllable at the end of sentence.

Current speaker—describing the current speaker (not set if the floor is available, i. e., nobody is claiming the speaking turn).

Current addressees—describing the current addressees as far as they can be inferred.

Current overhearers—describing the participants that are classified as overhearers.

Current bystanders—describing the participants that are classified as bystanders.

Previous speaker—holding the participant that was the speaker in the previous turn.

Previous addressees—holding the participants that were addressed in the previous turn.

The kind of information represented in the conversational context also depends on availability issues (e. g., in some cases the sensoric devices are not yet able to capture the information) and for some system configurations it simply does not make sense to model the information (e. g., in a classic dialogue system there are only two participants, the user and the system, which makes the representation of bystanders and over-hearers obsolete). Thus, the conversational status is not a fixed model but rather flexible with respect to the needs and capabilities of the dialogue system.

6.2.3 Modeling the Physical Environment

A key aspect of situated multimodal communication is the need for a comprehensive representation of the physical context within which the interaction takes place. As we have discussed in section 3.2.3, people integrate both verbal and nonverbal references to the physical environment into their contributions. Consequently, a comprehensive context model needs to incorporate a detailed model of the physical environment, including all objects, persons and other entities that are in the visual field of the dialogue participants and that might be referenced during the discourse.

As we have discussed in section 3.2.3, it is also important to model the relations between the Physical Objects by means of differentiation criteria like size, color and shape. These differentiation criteria can be used to non-ambiguously identify a referent out of a number of similar referents.

Another key aspect of this model of a physical environment is that it supports the cascading of embedded physical environments. This feature enables the modeling of distinct areas that themselves represent a closed physical environment (e. g., lists that are displayed on a screen, or rooms that are part of a physical environment modeling a house or an apartment). Once an object of the physical environment has been referenced, the object will also appear in the discourse context (see below).

The concept of physical environments we have developed for this work is comparable to the concept of *domains of references* of Salmon-Alt (2000). However, it is more flexible since it supports a spatial representation of the objects located in the physical surroundings.

Representing Basic Spatial Relations

Representing spatial relations between objects in the physical environment has a long research tradition and a number of comprehensive approaches and models have been put forward (see for example Maass (1996); Gapp (1996); Blocher (1999); Fuhr et al. (1995)). In what follows, we will describe a model for representing spatial relations between Physical Objects that is tailored to the task of resolving spatial references for multimodal dialogue systems. This model is based on labeled spatial relations between objects but does not incorporate as many details as other approaches.

We model objects that are located within a closed physical environment by means of either six or ten spatial relations, *northernNeighbor*, *north-easternNeighbor*, *easternNeighbor*, *southernNeighbor*, *south-easternNeighbor*, *westernNeighbor*, *north-westernNeighbor*, *above* and *below* plus the current orientation of each object (*north*, *east*, *south* and *west*). Figure 6.1 gives a graphical representation of these two types of spatial models. The concept of labeled spatial relations resemble the *spatial prepositions* used in (Fuhr et al., 1995) (see section 4.2). Our model, however, has several advantages compared to their model. First, as we will see below, our model does not need to maintain two layers—one modeling the scene independent of reference frames and another one modeling it from the perspective of the currently activated frame of reference—but it only needs a single representation. Second, since we also model the current orientation of an object, we do not need to recompute the complete model if the frame of reference changes.

Our model is organized as follows: each object located in the scene is represented by means of an *AbsolutePosition* (see section 8.2). An *AbsolutePosition* is represented in our meta-ontology but for now it is sufficient to think of a typed feature structure that comprises a set of features. The most important features of this structure are:

- Feature *ontologicalInstance*—this feature contains the ontological instance representing the object that is described by an *AbsolutePosition*.
- Feature *coordinates*—this is an optional feature: If the perception components are able to provide two-dimensional or three-dimensional coordinates describing the location of the object with respect to a fixed coordinate system, this slot will contain that information.
- Feature *orientation*—this feature describes the current orientation of the object; valid values are: *north*, *east*, *south*, *west*.
- Feature *northernNeighbor*—this feature contains a link to the *AbsolutePosition* of its northern neighbors (can be empty if there is no northern neighbor).

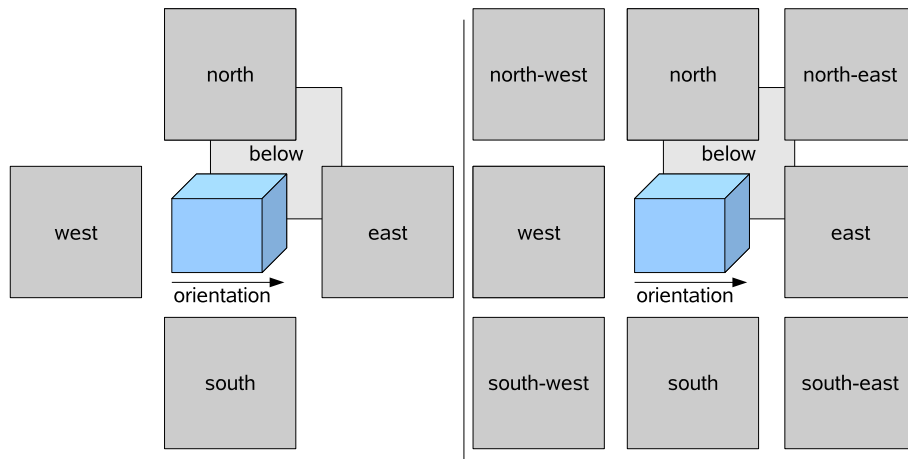


Figure 6.1: Graphical representation of the spatial relations between an object of the physical environment and its neighboring objects. The left part of the figure shows the coarse-grained model using six relations and the right part shows the fine-grained model using ten relations. Due to layout restrictions, both figures do not show the *above* relation.

- Feature *easternNeighbor*—this feature contains a link to the *AbsolutePosition* of its eastern neighbors (can be empty if there is no eastern neighbor).
- Feature *southernNeighbor*—this feature contains a link to the *AbsolutePosition* of its southern neighbors (can be empty if there is no southern neighbor).
- Feature *westernNeighbor*—this feature contains a link to the *AbsolutePosition* of its western neighbors (can be empty if there is no western neighbor).
- Feature *above*—this feature contains a link to the *AbsolutePosition* of its neighbors above (can be empty if there is no neighbor above the object).
- Feature *below*—this feature contains a link to the *AbsolutePosition* of its neighbors below (can be empty if there is no neighbor below the object).

This means that each *AbsolutePosition* that represents an object in the scene also represents the spatial relations the object currently fulfills with respect to the organization of the scene from an absolute point of view (using viewpoint neutral descriptions). If we consider, for example, a very simple physical environment consisting of a football team lineup, the idea of this representation will become clearer. Figure 6.2 depicts the spatial arrangement of the individual football players. Given this scene the representation of the physical environment looks like this:

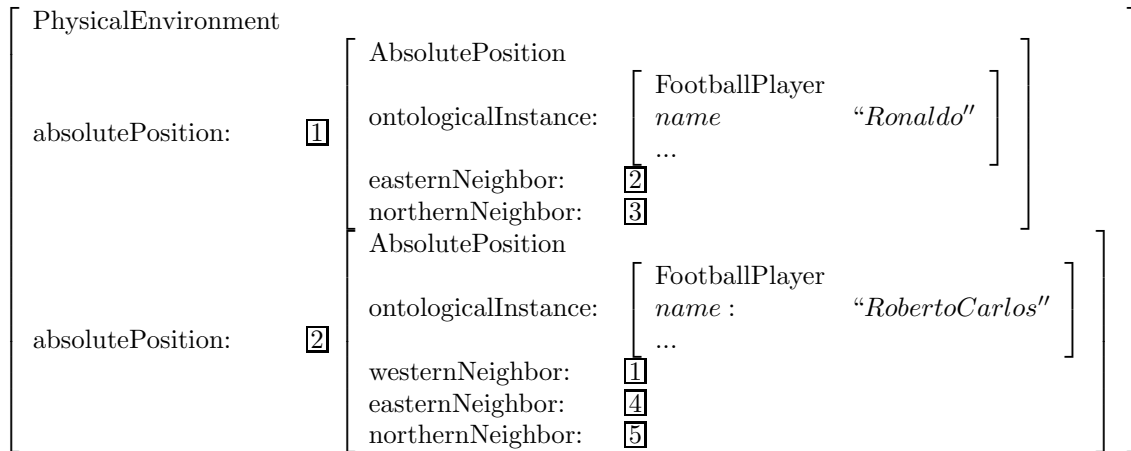


Figure 6.2: Example configuration of a physical environment: The football player Ronaldo is either to the right of Roberto Carlos or to the left of him, depending on the point of view.

Types of Spatial Organizations

Typically, the individual spatial organizations represent logical groups of objects that belong together either because of their spatial assembly or because of shared features like type, color and size. Examples of such groupings are the players on a football field or the elements of a list. There is virtually an unlimited number of possible organizations such as lists, double-column lists, set football fields, group photos and maps.

Each type of spatial organization comes with its own vocabulary that is typically used for place deixis or spatial references. The entries of a single-column list, for instance, can be referenced to by expressions like *the first*, *the second* or *the last*. However, some spatial organizations require more complex expressions, as for example references to a double-column list also require the determination of the row, e. g., *the first one on the right*. Other examples of complex spatial organizations are a football field or a map.

6.3 The Discourse Model

The purpose of the discourse model is to gain a comprehensive representation of the contributions to the propositional layer of an ongoing interaction to enable the resolution of referring and elliptical expressions. Given all the possibilities of referring expressions that are discussed in section 3.2.3, our goal is to design a context model that addresses the two central challenges of references resolution:

- 1 What information should the system maintain that might be useful for resolving future references?
- 2 Given an anaphoric reference in a follow-up sentence, how can we pick out the entity it is meant to represent?

The basis of our discourse model is the model we developed for the SMARTWEB system Pfleger (2002). Following the *three-tiered discourse representation* of Luperfoy (1991), our discourse model comprises two main layers: (i) a Modality Layer—extending her linguistic layer and (ii) a Referential Layer—extending her discourse layer. The third layer—the knowledge base or belief system—corresponds in our approach to the long-term memory. Besides this, the discourse context also comprises a sequential representation of the course of the interaction, i. e., a discourse history. This discourse history encompasses information about the speaker that performed the utterance and the propositional content. Formally, we define the discourse context DC_A of a participant A as a set of $\langle ML, RL, LTM, DH \rangle$, where ML corresponds to the Modality Layer, RL corresponds to the Referential Layer, LTM corresponds to the long-term memory, and DH to the discourse history.

In the remainder of this section, the first two layers of this discourse model will be discussed in detail. The long-term memory will be introduced in the next section.

6.3.1 Modality Layer

The objects of the Modality Layer provide information about the surface realizations of objects at the Referential Layer that have been introduced into the discourse. Thus, the Modality Layer describes the circumstances that caused the increase in activation of their corresponding instance. The Modality Layer consists of three classes of objects reflecting the modality by which the corresponding Referential Object was referenced: (i) *Linguistic Actions*, (ii) *Nonverbal Actions*, and (iii) *Physical Objects*.

Linguistic Actions

Linguistic Actions resemble the linguistic objects of Luperfoy (1991). They comprise information about the surface realization of an instance like lexical information (the lemma used to reference an instance), syntactical information (e. g., number, gender or case), its realization time or the type of reference (e. g., definite/indefinite, deictic/anaphoric/partial anaphoric). Each Linguistic Action is linked to exactly one instance of the knowledge base and when this link is established, the activation of the referenced object is increased. Linguistic Actions are of particular importance for the resolution of referring expressions as they provide the linguistic information needed to identify co-references on the linguistic level.

A Linguistic Action is described by the following features: (i) the point in time at which the referring expression was uttered, (ii) the lemma used to reference the corresponding instance,

(iii) the syntactical information (i. e., number, gender and case of the referring expression), (iv) the type of the reference (i. e., definite, indefinite, deictic, anaphoric, or partial) and (v) the participant that realized the action.

Nonverbal Actions

Nonverbal Actions represent the nonverbal behavior of the interlocutors that contribute to the propositional content of the utterance (e. g., pointing gestures, iconic gestures, emblematic gestures, but also gaze behavior or drawings). Nonverbal Actions comprise information about the type of nonverbal action, its start and ending time. Nonverbal Actions facilitate the resolution of deictic expressions (e. g., “*What’s the name of that [pointing gesture] player?*”).

A Nonverbal Action is described by the following features: (i) the point in time at which the nonverbal action was displayed, (ii) the duration during which the nonverbal action was displayed, (iii) an ontological representation of the nonverbal action (e. g., deictic, iconic, emblematic, gaze), (iv) a link to the referenced object at the Referential Layer and (v) the participant that realized the action.

Physical Actions

Physical Actions describe changes of the state of the physical world (e. g. the appearance or disappearance of objects in the physical environment or intentional actions) performed by a participant of the discourse (see section 2.3.7). They comprise information about the type of the event, when it happened, and about the spatial properties of that object (including its relative position to other objects in the scene).

A Physical Action is described by the following features: (i) its realization time, i. e., the time point when a Physical Action began, (ii) the duration of the Physical Action, (iii) an ontological representation of the Physical Object at the Referential Layer that is involved in the Physical Action, (iv) an ontological representation of the type of the physical action (e. g., *Appear, Disappear*), (v) a representation of the location of that object represented by means of an absolute position (see section 6.2.3) and (vi) the participant that realized the action.

6.3.2 Referential Layer

Objects at the Referential Layer provide the link to the instances of the long-term memory (see section 6.4). Each object at the Referential Layer (if completely disambiguated) represents a unique instance of the long-term memory whose activation value exceeds the threshold that differentiates between activated and in-activated objects (see section 6.4). We distinguish three types of objects at the Referential Layer: (i) Discourse Objects, (ii) Implicitly Activated Objects and (iii) Physical Objects.

6.3.3 Discourse Objects

Discourse Objects are containers for instances that were directly mentioned during the preceding discourse. They comprise a unified representation of the semantic information gathered so far. In case a Discourse Object is completely resolved, its unified representation is replaced by a link to the corresponding instance of the long-term memory. Additionally, it contains a set of links to instances of the long-term memory (more than one in case of

ambiguous/under-specified references) and links to objects at the Modality Layer; every time a Discourse Object is mentioned, a new link is added.

Implicitly Activated Objects

Implicitly Activated Objects are objects that are activated by means of the mentioning of a Discourse Object that is associated with them (e. g., instances like the pilot and the passengers in the context of the mentioning of a plane). If a Discourse Object accesses an instance in the LTM, the activation of instances related to it is increased by a dynamic factor which depends on the activation of the superordinated instance and the *strength* of the relation between them. The spreading of activation is a recursive process (see section 6.4).

Implicitly Activated Objects may appear in the discourse context when their corresponding Discourse Object appears. This happens in case their activation exceeds the threshold. Objects are directly connected to this. Also, the activation of Implicitly Activated Objects decreases faster than that of Discourse Objects. Consequently, they are only accessible for a short time.

Physical Objects

Physical Objects represent objects that can be perceived from the visual environment. If a Physical Object is explicitly activated through the mentioning of a Discourse Object, it can serve as a referent for a referring expression. Physical Objects are not only part of the Referential Layer as they are also part of a superordinate structure representing the complete physical surroundings by modeling the relations between the Physical Objects located in a scene (e. g., the grey building is to the left of the blue building).

6.4 The Long-Term Memory

A key feature of our context model is the existence of a long-term memory (LTM) that is connected with the working memory (in psychology it is currently also common to view the working memory as a distinct part of the LTM). This memory unit provides access to all available but not directly accessible knowledge of the dialogue system. The objects represented in our LTM form a semantic network where the individual instances are connected through relations (i. e., the properties of the ontological classes). The left part of figure 6.3 depicts a small excerpt of such a semantic network. At the bottom of this figure there is an instance representing the German football player *Michael Ballack*. This instance exhibits several connections with other instances like teammates, or the German national team that participated in the game against the USA at the World Cup in 2002.

An important aspect of this LTM is that every object has an activation value defining its accessibility. The higher the activation value, the easier it is to access the object (i. e., to retrieve the object from the complete set of knowledge). To account for the activation of neighboring instances that can be observed in human interactions, the activation of a knowledge chunk is passed on to its associated chunks by a process called *spreading activation*. Spreading activation means not only that each connected object receives part of the activation of its neighbors, but also that it spreads its own activation on to its neighbors.

The task of the LTM within this context model is to provide access to the actual referents for referring expressions including their associated entities. This information can be used for

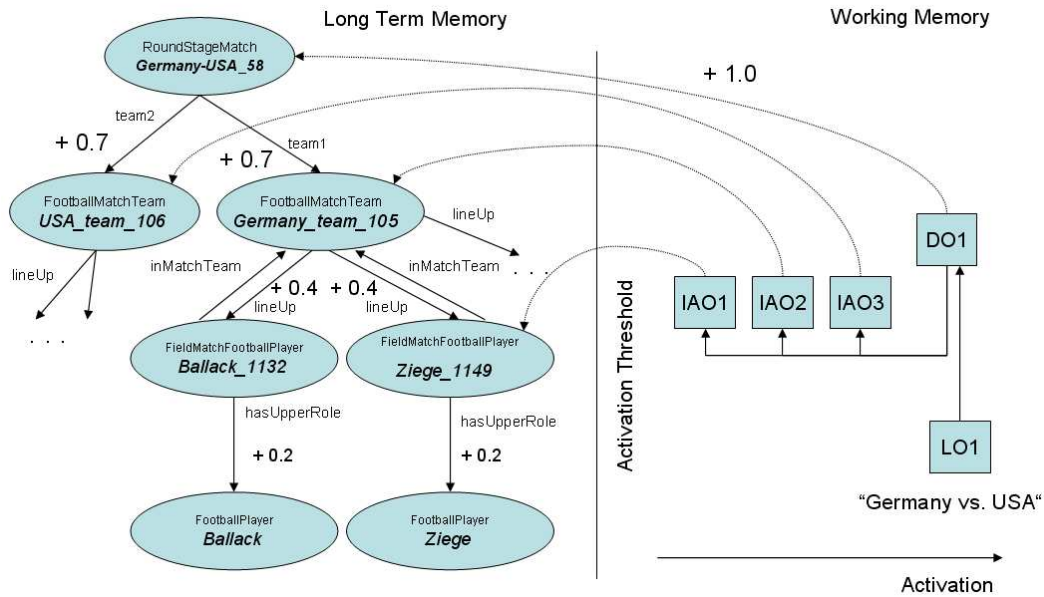


Figure 6.3: Basic organization of the context model (taken from (Pfleger and Alexandersson, 2006)): The left part of the figure shows an excerpt of the long-term memory and the right part shows some objects on the Referential Layer in the working memory. Both memory structures are separated by an activation threshold.

the resolution of implicit referring expressions (see section 3.2.1).

All entities of the long-term memory are first not accessible, i.e., they are not directly accessible by means of referring expressions. The status of an inactivated entity only changes if its activation exceeds a certain threshold that marks the border between accessible and inaccessible entities. All entities whose activation is below this activation threshold are not accessible for the reference resolution process. However, if instances receive additional activation by means of the spreading activation process, they can exceed the activation threshold which makes them accessible for the reference resolution process.

6.5 The Context Changing Function of Utterances, Physical Actions and Events

As discussed in section 2.2.5, every communicative or physical action that is uttered within the course of an interaction has to be interpreted in the light of its context but also changes this context. Formally, this is realized by means of a context update function $F(p,i,s,a)$ that takes the propositional content p , the interactional content i , the speaker s , and addresses a of an utterance or a physical action and updates a given context Γ to Γ' . In the following two subsections, we will discuss how this is achieved for the immediate conversational context and discourse context, respectively.

Besides utterances and physical actions, we distinguish a set of four basic events that also have a context changing function: (i) *Join(a)*—a participant a joins the conversation, (ii)

Leave(a)—a participant a leaves the conversation, (iii) *StartOfTurn*—a participant takes the turn and (iv) *EndOfTurn*—a participants ends a turn. These basic events are also incorporated into the context by means of four context update functions *Join*, *Leave*, *StartOfTurn* and *EndOfTurn* that all take an agent a as argument in order to update a given context Γ to Γ' .

6.5.1 Updating the Immediate Conversational Context

As discussed in section 6.2, the conversational context IC_A of a participant A is represented by the set $\langle CS, P_i, \dots, P_n \rangle$, where CS is the conversational status and P_i, \dots, P_n are the registered participants. There are two primary update functions for dealing with changes in the number of present participants. The context update function *Join(a)* updates the list of registered participants by adding the participant a . Accordingly, the function *Leave(a)* removes the participant a from the list of registered participants. Additionally, the context update function *StartOfTurn(a)* registers the specified agent a as the current speaker of the interaction while the function *EndOfTurn(a)* changes a 's status from current speaker to previous speaker.

When a participant A performs a nonverbal action NA , the immediate conversational context is changed so that the representation of that participant (e. g., P_A) will comprise the performed nonverbal action in the field *Nonverbal behavior* for the duration of the nonverbal action.

6.5.2 Updating the Discourse Context

The discourse context is defined as the set $\langle ML, RL, LTM, DH \rangle$, where ML corresponds to the Modality Layer, RL corresponds to the Referential Layer, LTM corresponds to the long-term memory, and DH to the discourse history. When a participant P_B performs an utterance, all four aspects of the discourse context have to be updated. The discourse history is updated by adding the propositional content together with information about the speaker at the end of the sequence. The other three layers are extended as follows: First, the propositional content p has to be analyzed with respect to the individual Referential Objects that are referenced in p^2 . For each referenced Referential Object it is first determined whether it has already been mentioned during the previous discourse. If this is not the case, the Referential Layer will be extended with this Referential Object. Otherwise, the existing Referential Object will be used to link it with the corresponding Modality Object (that is either a Linguistic Object in case of a verbal reference, a Gestural Object in case of a gestural reference or a Physical Action in case of a physical action).

In a second step, the activations of the objects at the Referential Layer and in the long-term memory that have been mentioned during a turn are increased accordingly. In contrast, the activation of all Discourse Objects at the Referential Layer that have not been mentioned in the current utterance will be reduced and all objects whose activation is below 0 will be removed from the Referential Layer, which means that they are no longer accessible. Finally, the spreading activation process of the long-term memory ensures that the Implicitly Activated Objects at the Referential Layer will be updated.

²Note that a Referential Object can be referenced by means of both verbal and nonverbal actions.

6.6 Summary

In this chapter we have presented a comprehensive context model for multimodal multiparty discourse. At the top-level, this context model differentiates between an immediate conversational context and a discourse context. As discussed in Chapter 2, it is important for the interpretation of interactional contributions to consider the immediate status of the conversation and the current conversational role of the participants which are provided by the immediate conversational context. The discourse context realizes a discourse history consisting of the individual propositional contributions of the participants. This discourse history centers around a representation of the individual referents of a discourse and comprises not only verbally mentioned instances, but also those that are introduced through nonverbal actions.

The second key feature of this context model is the associative long-term memory. A long-term memory is usually not considered to be part of a classic context model. Usually, as in the work of Luperfoy (1991), there is no more than a link or direct representation of some knowledge base entries that are considered on their own. However, as we have seen in the previous sections, some processes in the human LTM have a direct impact on the organization and structure of the contextual model. To this end, our approach integrates a structure resembling the human long-term memory. Moreover the activation spreading within the long-term memory ensures that not only the explicitly mentioned knowledge is accessible but also associated knowledge. We consider this to be the primary key for the resolution of implicit references (see section 3.2.1).

Chapter 7

PATE—A Production Rule System Based on Typed Feature Structures

In this chapter we discuss the central aspects of the production rule system PATE. Production rule systems in general provide easy means to state rules—consisting of a condition and an action part—that are constantly applied to a working memory. The working memory is a controlled representation of the objects that describe a situation in the world of interest. Rule application is done by first testing all rules as to whether their conditions are fulfilled by the current configuration of the working memory. The production rule system then selects a rule to fire out of the set of applicable rules and applies its actions to the working memory. This might change the current configuration of the working memory. In section 7.1 we first give a brief overview of production rule systems in general and of the development of PATE. In section 7.2 we discuss the PATE’s architecture before in section 7.3 the syntax of production rules is discussed. Section 7.4 outlines the overall processing strategy and section 7.5 gives an overview of PATE’s graphical user interface. Section 7.6 then discusses the internal data representation of PATE.

7.1 Introduction

A multimodal fusion and discourse processing component has to deal with a great variety of input patterns. Some of these patterns require reactive processing (i. e., interactional contributions) with minimal delays, while the rest require a more complex processing (i. e., propositional contributions). Moreover, the processing logic for dealing with interactional contributions and controlling the reference resolution process can best be expressed by means of rules that match against the current configuration of the conversational context and then execute appropriate actions. These two characteristics of the task at hand led to the decision to use a production rule system for the main control loop of FADE.

The basic idea of a production rule system is the invocation of procedures or methods based on the matching of a pattern. This *pattern matching* based processing stands in contrast to the hierarchical control of popular computational languages like C, C++ or Java. In these languages, the control is usually based on moving from one function or method to the next, thereby executing a fixed sequence of individual commands expressed in the body of these functions. This kind of processing is very effective for clear-cut problems whose solution can be laid out in advance. However, the problems that have to be solved when developing a

dialogue system are often far from being clear-cut because of their great variety of input events.

In what follows, we will introduce our implementation of a production rule system called PATE¹ which is specifically suited for the development of applications that are embedded in multimodal dialogue systems. The main focus of PATE is to provide a component that can be used to organize the main processing loop of FADE. However, because of its generic design, PATE is also a toolkit that ideally supports the rapid development of applications for multimodal dialogue systems.

7.1.1 Development of PATE

In the first version of the PATE system we introduced an extended production rule system for the task of multimodal fusion (see Pflieger (2004); Kempe (2004)). The system incorporated central aspects of the ACT-R 4.0 system of Anderson and Lebiere (1998). The general idea is that all incoming data will be assigned an activation value before it is stored in a *working memory* (WM). This activation value represents the current accessibility of a *working memory element* (WME) and fades out in time. If the activation of a WME exceeds a specific threshold, this particular WME is not accessible anymore. The data stored within a WME is represented in terms of the typed feature structures (see section 7.6.2) which allow for an easy conversion of many different data representation frameworks, as for example ontology based ones. Parallel to the working memory there is also a so-called goal-stack (see Anderson and Lebiere (1998)) which defines the focus of attention of the system—the WME on top of the goal-stack defines the current focus of the PATE system. Even though Anderson and Lebiere abandoned the goal-stack in later versions of the ACT-R system, PATE still supports this feature.

The next iteration of the PATE system (PATE 4.0; see Pflieger and Schehl (2006)) introduced improvements focusing on the overall performance of PATE. Eventually, we were able to reduce the overall processing time for a single processing cycle by a factor of three. This has been achieved, for example, by a reimplementing of the underlying data representation (see section 7.6). Additionally, we added the support of plug-ins for conditions (see section 7.3.1) and improved the internal knowledge representation. The current version of PATE (PATE 5.0) also features a long-term memory that is directly linked with the working memory and supports the processing of large knowledge bases (e. g., more than 60k instances). We also added several extensions to both the condition and the action part based on the feedback we received from people that have been developing applications with PATE.

7.1.2 Extending Classic Production Rule Systems

In the past twenty years, several production rule systems have been put forward. While the core processing mechanism is the same for most of the systems, they differ in aspects like data representation and expressiveness of the pattern language and action parts. Similar to a traditional production rule system, there is a set of production rules that can be applied to the objects stored in the working memory. Each production rule consists of three components: (i) a weighting, (ii) a condition part and (iii) an action part. The weighting will be used during conflict resolution to compute an overall score for a rule, but we will return to this later. For the sake of writing and maintaining production rules, the PATE system also includes a

¹PATE stands for *A Production rule system based on Activation and Typed feature structure Elements*.

built-in graphical user interface. Another important aspect of the PATE system is its easy adaptation to new tasks or entirely new dialogue systems. All relevant information is stored in a global configuration file containing general definitions of system-wide variables, a file defining the type system and at least one file defining the production rules. Finally, PATE also encompasses a set of XSLT stylesheets that manage the conversion of in- and out-going data for communicating with external components (if necessary).

7.1.3 Related Work

Today, there are many practical implementations of production systems like SOAR (see Laird et al. (1987)), ACT-R (see Anderson and Lebiere (1998)), CLIPS (*C Language Integrated Production System*; see <http://www.ghg.net/clips/CLIPS.html>), Jess (see <http://herzberg.ca.sandia.gov/>), or JEOPS (see da Figueira Filho and Ramalho (2000)) which raises the question of why we have decided to implement our own production system. The short answer to this question is that the existing approaches do not completely satisfy the requirements of multimodal dialogue systems.

As mentioned in the previous section, all production rule systems realize more or less the same processing logic: The condition parts of the rules specified in the rule base are applied to some kind of a working memory which results in a conflict set comprising one or more rules that can potentially fire. Out of this rule set, the system selects a rule to fire. Then, the action part of the rule will be executed which possibly changes the configuration of the working memory. Up to this point, there is no big difference between the individual systems. However, they vary with respect to data representation, their expressiveness in the condition and action part, and the way the conflict set is computed and the firing rule is selected. We will start with a brief requirements analysis and then we will compare PATE with the existing approaches.

7.1.4 Requirement Analysis

With respect to the intended use of the production system as a generic toolkit for developing components of a multimodal dialogue system, we have identified the following requirements for PATE:

Portability/programming language: The production system should run under Linux as well as Windows and should be implemented in Java since this is the programming language that is used for all target systems. An integration of a production system that is implemented in another programming language has been ruled out since the necessary serialization and parsing of data would be too time consuming.

Support of ontological data representation: Since the data in the target system is represented by means of ontologies, it is a major requirement that the production system supports a direct processing of ontological data without the need of complex data conversion.

Support of unification and overlay: As discussed in section 5.2.4, unification and overlay are two central operations for comparing and manipulating ontological data. These two operations should therefore be provided by the production system for condition matching and data manipulation.

Support of a plug-in mechanism: It should be possible to integrate Java plug-ins into the production system that can be used both in the condition part and the action part of production rules. Such a plug-in can be used for complex computations that cannot directly be realized using the supported rule language.

Activation-based data representation: In order to be able to simulate fade-out effects of activated knowledge chunks, the production system should support an activation-based data representation.

Comparing PATE to Existing Approaches

When we consider the first requirement, it follows that most of the aforementioned existing approaches have to be ruled out. Only Jess and JEOPS are implemented in Java; all other approaches are either realized in Lisp or in C/C++.

Another important aspect for us is the way in which the content data and the patterns of the condition and action part are represented. As discussed in chapter 9, our system context is based on ontological data represented in diverse formats. In order to prevent time-consuming conversions of incoming and outgoing data, we need a direct support of ontological data. In addition to that, the production system should make use of the taxonomy provided by the ontology, including complex nested structures and subsumption relations for pattern matching. CLIPS, for example, supports the definition of a type hierarchy including subsumption relations, but its Lisp-like representation of data makes it difficult to deal with complex nested structures as they occur in multimodal dialogue systems. Jess is a Java port of CLIPS and shows the same Lisp-like syntax and does not support an activation-based data representation. Neither CLIPS nor Jess support unification and overlay and thus would require significant extensions. JEOPS follows a different approach by extending the standard Java language with the expressiveness of a production system. However, in this approach the data is represented and manipulated by means of fields and methods of the Java language which makes the process of coding rules a programming task rather than a declarative specification of patterns. ACT-R, in contrast, fulfills most of the specified requirements, however it does not support unification and overlay, and it is implemented in LISP.

To conclude, none of the existing approaches completely fulfills the specified requirements. As a consequence, we designed and implemented PATE. In the following sections we will provide an overview of PATE and its underlying knowledge representation.

7.2 The Architecture of PATE

PATE's architecture is centered around the idea of three separated data storage facilities: (i) the *goal-stack*, (ii) the *working memory* and (iii) the *long-term memory*. While the working memory holds the activated instances that are accessible for processing and rule application, the long-term memory serves as a kind of persistent storage for all the instances the system is aware of. The goal-stack contains distinct instances that are used to control the application of the rules. Figure 7.1 shows the basic architecture of PATE including the abstract processing steps. In the following we will first discuss the roles of the three data storages before we will discuss the fetch-execute cycle which is the central processing loop of PATE.

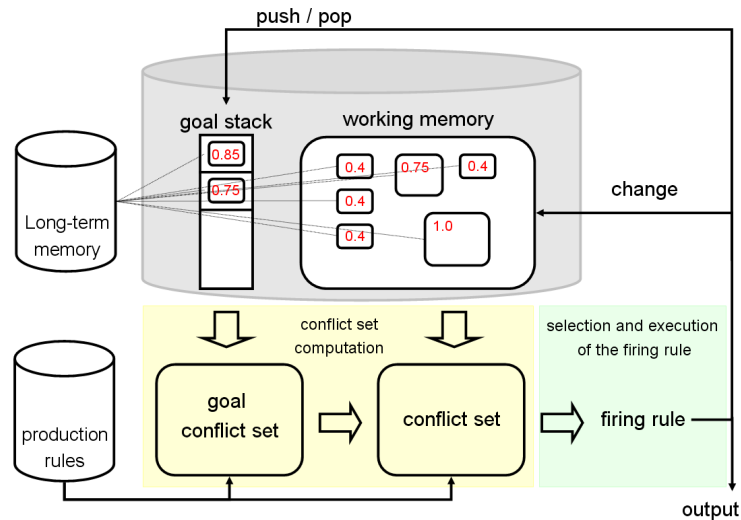


Figure 7.1: Basic architecture of PATE. The system comprises three interweaved memory areas: (i) a working memory, (ii) a goal-stack and (iii) a long-term memory. The applicability of the production rules is tested in two steps: First, a goal conflict set is computed that encompasses all rules whose goal condition is fulfilled. Then, based on this preselection, the actual conflict set is determined. The highest ranked rule is selected to fire which possibly results in changes to the working memory and the goal-stack.

7.2.1 The Working Memory

The working memory of PATE serves as the central memory for all activated instances that can be used for rule application. The objects located in the working memory are called *Working Memory Elements* (WMEs). A WME comprises information about its creation time and its activation. The content of a WME is an instance of an eTFS object, PATE's internal format for data representation (see section 7.6). All WMEs whose activation value falls below a certain threshold are not accessible anymore and will be removed from the working memory.

7.2.2 The Goal-Stack

The idea of employing a goal-stack in a production rule system originates from (Anderson and Lebiere, 1998). The purpose of this goal-stack is to model the focus of attention of the system (Anderson and Lebiere, 1998) so that it will always have a single WME in focus. This focused WME has a direct impact on the number of rules that need to be considered for conflict set computation (see section 7.4). Only the rules whose goal-condition are fulfilled by the WME on top of the goal-stack have a chance to fire.

The goal-stack is closely interwoven with the working memory, as a WME that is accessible can only exist in one of these two memory areas. A WME that has previously been located in the working memory can be added to the top of to the goal-stack by a *push* operation but this has the side-effect that the WME disappears from the working memory. On the other hand, the *pop* operation removes an object from the goal-stack and adds it to the working memory.

7.2.3 The Long-Term Memory

The long-term memory (LTM) realizes an efficient repository for storing all kinds of knowledge the system is aware of but which is not directly accessible. This knowledge is represented in terms of instances of eTFS objects. Each eTFS object of the LTM has an activation value that is updated via a spreading activation mechanism (see section 7.6). If the activation of a WME exceeds a certain threshold, the object appears in the working memory, i. e., it is accessible for rule application. In contrast to the working memory and the goal-stack, the LTM is optional, which means that it can be used but that it is not required. As a consequence, the application of the spreading activation mechanism must be triggered manually by calling a specific plug-in that takes the WMEs that serve as the starting point for the activation spreading as arguments (see section 7.2.3).

The Organization of the Long-Term Memory

Since the LTM of PATE is the central repository for knowledge, it is likely that it will contain a great number of instances of eTFS objects. Furthermore, the main purpose of the LTM is to provide efficient search operations that support the retrieval via unique identifiers and via partially instantiated search patterns using unification. Especially in the latter case, a simple search through all available instances would be extremely inefficient, as the instance base of an ontology can easily exceed 50K instances².

This means that the organization of the LTM should already support the retrieval. To this end, we developed an internal organization of the LTM that is based on the type-hierarchy and on the activation of the instances. We use the type-hierarchy to partition the long-term memory into a number of sub-areas where all instances of the same type are stored. These sub-partitions in turn are organized by means of the activation of the individual instances: The higher the activation, the further ahead the instances appear. This kind of storage drastically reduces the search space during the retrieval of objects from the LTM. A performance test with the SMARTWEB ontology (see section 5.4.2), which comprised at that time more than 60K instances, revealed a considerably better retrieval time compared to a simple search in the complete instance base.

Lookup and Retrieval of Instances from the LTM

The instances of the LTM are stored directly in the type-hierarchy, i. e., every type also provides a storage position for its instances. This means a instance of type *A* is stored directly at that type and can be retrieved in turn via that type. Thus, lookup and retrieval of instances of the LTM is comparatively cheap, as the type of the search pattern already restricts the search space to a restricted subset of the complete knowledge base. The actual matching is done by unifying the search pattern with the individual instances stored for that type. Note that since the lookup is typically based on an under-specified instance, it might return more than one match. In such a case, the process that initiated the lookup must deal with this ambiguity.

The PATE API provides three methods for the retrieval of LTM instances. The first is called *retrieveTFSbyID* and takes a String as argument (e. g, a *Uniform Resource Identifier* (URI)³). This method looks up the object associated with the specified id and returns it if

²The current SMARTWEB ontology, for example, comprises more than 60K instances.

³See <http://tools.ietf.org/html/rfc3986>

it exists. Additionally, there is a retrieval method called *retrieveTFS* that takes an under-specified search pattern as arguments and returns all objects of the LTM that unify with that pattern. Besides these retrieval methods there is another way to transfer instances of the LTM to the working memory. If the activation of a instance exceeds a specific threshold, it becomes accessible in PATE's working memory. Note that this does not mean that the instance is copied or moved from the LTM to the working memory. In fact, both memories will provide access to the same instance in that case. Consequently, if the instance is altered in the working memory, this has the side-effect that the object in the LTM is also altered. But it is also possible to directly access the LTM from a production rule using a built-in plug-in called `RetrieveInstancesFromLTM` (see section 7.3.1 for an overview of PATE's plug-in mechanism). This plug-in takes a WME as argument and adds all instances from the LTM to the working memory that are unifiable with that argument.

Spreading of Activation

In the following we will briefly describe the spreading of activation within PATE's LTM, a detailed description of the complete approach and its implementation can be found in (Gehrmann, 2007). An important aspect of our LTM is that every object has an activation value defining its accessibility. The higher the activation value, the easier it is to access the object (i. e., to retrieve the object from the complete set of knowledge). To account for the activation of neighboring instances that can be observed in human interactions, the activation of a knowledge chunk is passed on to its associated chunks by a process called *spreading activation*. Spreading activation means not only that each connected object receives part of the activation of its neighbors, but also that it spreads its own activation on to its neighbors. The idea of using spreading activation in semantic networks originates from cognitive psychology (see Collins and Loftus (1975); Anderson (1983)).

Priority Relations

Usually, an instance of an ontological class encompasses a great number of slots and these are not always equally relevant. Consider, for example, an instance describing a football team. Here, the coach and the players are more important and should receive more activation than the groundskeeper or the bus driver. As discussed in Gehrmann (2007), activation spreading in PATE not only takes place by traversing all filled slots of the individual instances, but also by means of *Priority Relations* (PR). A Priority Relation provides a means to realize primary associations between an instance of a particular class and one or more of its slots. Figure 7.2 shows an example of PRs as they are used for the SMARTWEB system. Besides the individual slots, it is also possible to define whether a PR can always be applied or only at certain points in the activation spreading process. The attribute *root* specifies that the PR can only be applied at the beginning of the spreading activation (i. e., if the current instance is the starting node). Otherwise the PR would not be applicable. The value *default* specifies that the PR is always applicable. See Gehrmann (2007) for more details on Priority Relations.

Activation Decay

In order to prevent an over-activation of the long-term memory, the activation of an instance fades out in time. This means that the longer an instance has not been referenced, the lower its activation will be. However, the activation of an instance will never get below

```

<priorityRelations4Type type="sportevent#Team">
  <priorityRelation context="default" path="sportevent#topScorer"/>
  <priorityRelation context="root" path="sportevent#topScorer"/>
</priorityRelations4Type>
<priorityRelations4Type type="sportevent#FootballTeam">
  <priorityRelation context="default" path="sportevent#partOf"/>
  <priorityRelation context="root" path="sportevent#partOf"/>
  <priorityRelation context="root" path="sportevent#inTournament"/>
</priorityRelations4Type>
<priorityRelations4Type type="sportevent#MatchTeam">
  <priorityRelation context="default" path="sportevent#bench"/>
  <priorityRelation context="default" path="sportevent#lineup"/>
  <priorityRelation context="root" path="sportevent#bench"/>
  <priorityRelation context="root" path="sportevent#lineup"/>
  <priorityRelation context="root" path="sportevent#inMatch"/>
</priorityRelations4Type>

```

Figure 7.2: Example of Priority Relations for defining primary associations between an instance of a class and a subset of its slots (from Gehrmann (2007)).

its basic activation. Eventually, when the instance's activation is below the threshold, the corresponding object at the Referential Layer (see section 6.3.2) of the working memory will disappear so that the object is no longer directly accessible. The three different types of objects at the Referential Layer exhibit different intensities in activation decay. Implicitly Activated Objects show the most rapid decay, followed by Discourse Objects. The activation of a Physical Object normally remains at a level that is above the threshold where objects disappear from the working memory.

7.3 Production Rules

In this section we introduce the rule language of PATE and give an overview of the supported operations for the pattern matching and action part of the production rules. A production rule of the PATE system consists of three parts: (i) a *weighting* defining the initial accessibility of the rule, (ii) a *condition* part and (iii) an *action* part. The weighting of a rule changes over the course of processing. If a rule fires, its weighting will be decreased by a constant factor, otherwise its weighting will be increased. However, the weighting of a rule can never exceed its initial value. This rule-weighting mechanism ensures that a rule is prevented from firing constantly since at some point the rule's weighting falls below 0 which means that it is excluded from the conflict set (see section 7.4). The basic body of a production rule looks like this:

```

<rule name="sampleRule">
  <weighting> 1.0 </weighting>
  <comments> some example rule </comments>
  <conditions>
    <condition name="goal">
      ...
    </condition>
  </conditions>
  <actions>
    <action type="...">
      </action>
  </actions>
</rule>

```

7.3.1 Condition Part

Conditions form the restrictions that a configuration of a working memory must meet so that a rule can fire (i. e., it can be applied). The condition part itself consists of a set of conditions. The number of conditions is not restricted. However, the first condition is the so-called goal condition that is matched against the goal-stack. A condition defines a pattern that must be matched by an object of the working memory unless the condition is true. A condition that only matches if a WME of the type “Song” and the title “One” is in the working memory looks like this:

```
<condition name="varName1">
  <object type="Song">
    <slot name="title"> One </slot>
  </object>
</condition>
```

There are three operations available for pattern matching: (i) a restricted version of unification, (ii) unification and (iii) overlay (selected via the attribute *method* in the definition of the condition). If the pattern matching operation returns a result, it is bound to the variable defining the condition; otherwise the condition is false and the whole rule is removed from the conflict set.

```
<condition name="varName2" method="overlay">
  ...
</condition>
```

In contrast to conditions which are only true if there is an object in the WM that fulfills them, there are *not-conditions* that are only true if there is no object in the WM that fulfills them:

```
<not-condition name="varName2">
  ...
</not-condition>
```

The patterns defined by the condition can additionally contain further variable definitions that select parts of the matched WME. A variable can be used to reference parts of bound objects in the action part (see subsection 7.3.2). But if an already bound variable is referenced again, its content will be replaced by the unification of its content and the same substructure of the WME matched by the current condition. Each condition introduces at least one variable (defined by the name of the condition) to which the WME matching the condition is bound.

An additional class of conditions is introduced by temporal restrictions, e. g., *before* or *sameTime* which define temporal relations between two WMEs that were bound in previous conditions. A temporal restriction compares the creation times of the two WMEs and returns true if the temporal relation is fulfilled, or false if otherwise. However, the *sameTime* restriction tests whether the two WMEs are within the same time interval.

Plug-ins

Another feature of the condition part of PATE is the possibility to integrate additional plug-ins. To provide a general solution for problems that cannot be solved by means of standard conditions, a mechanism in PATE allows for arbitrary user-defined functions that can take arguments as WMEs and store the result in a new WME. In the condition part of a rule, the following syntax is used to get the result of a plug-in:

```
<function name=" text " arg1=" text " (arg2=" text "
      (arg3=" text " [...] )?)? result=" text " />
```

The name of the plug-in refers to a Java class of the same name residing in the path that has been set using the configuration variable *pluginDirectory*. This class extends the abstract class `de.dfki.productionSystem.Function` and has to implement the calculating method *applyFunction*. The arguments *arg1* to *arg10* are variable names that (if used) must have been bound in earlier conditions. In conditions and actions succeeding the function, the name specified by the attribute *result* can be used like a variable. It is also possible to use plug-ins in the action part (see the next section) of a rule. Here, plug-ins can be used to perform more complex operations that would be difficult or impossible to realize by means of standard actions.

7.3.2 Action Part

When a production rule is selected to fire, its action part is executed. This means that the defined actions are applied subsequentially to the current configuration of the goal-stack and the working memory. Every action—except for *output*—changes the current configuration of the working memory to some extent. The actual change of the working memory that is caused by a rule is also important for scoring the rule during the conflict resolution process (see section 7.4.1). Therefore, each action type contributes to the overall score of the action part with an individual weighting that is defined in the general configuration file of the PATE system. There are seven different actions in total that can be used within the action part of a rule. Some of them take no argument, e. g., *pop*, whereas some need a variable that determines its argument, e. g., *updateWME* or *push*. PATE currently supports the following actions:

updateWME: This action updates an existing WME (referenced through the attribute *name*) by means of a typed feature structure defined within the body of the action. This typed feature structure can be enriched through variables. The actual update of the referenced WME takes place through an overlay of that WME over the specified typed feature structure. Below, we give an example of an updateWME action. Suppose that the variable `myCar` was previously bound to an instance of a car that is green and the variable `Mary'sFavouriteColor` was bound to an instance of the color red. If we apply this action, the result is that the color of the instance `myCar` has now changed to red.

```
<action type="updateWME" name="myCar">
  <object type="Car">
    <slot name="has_color">
      <variable name="Mary'sFavouriteColor"/>
    </slot>
  </object>
</action>
```

addWME: This action does basically the same as the action updateWME, however, it adds the WME to the working memory if it does not exist in the WM.

```
<action type="addWME" name="myCar">
  <object type="Car">
    <slot name="has_color">
      <variable name="Mary'sFavouriteColor"/>
    </slot>
  </object>
</action>
```

pop: This action removes the topmost WME from the goal-stack and adds it to the working memory.

```
<action type="pop"/>
```

push: This action puts a WME on top of the goal-stack and removes it from the working memory.

```
<action type="push name="wme1"/>
```

deleteWME: Removes a WME either from the goal-stack or from the working-memory. Note that in contrast to the pop operation, a WME stemming from the goal-stack will not be added to the working memory.

```
<action type="deleteWME" name="wme1"/>
```

overlay/unification: Unification and overlay (see section 7.6.2) are used implicitly in condition matching, variable aggregation and certain other actions. However, when merging eTFS structures that are bound to variables, it is useful to be able to explicitly call the unification or overlay operations in the action part of a rule. PATE supports two action types to perform these calls: `unify` and `overlay` (see section 7.6.7). The operands are declared as to be variables within the action tags in order of their appliance (*operand1* is overlaid with *operand2*). The result of the operation is stored in a variable using the name of the action. If the unification fails (overlay always returns a result), an empty variable is created. If the first operand is an eTFS structure, the empty variable will be an empty eTFS structure, i. e., an eTFS of the root type in the type-hierarchy. If the operand is bound to a string value, an empty string value is returned.

```
<action type="overlay" name="result">
  <variable name="this"/>
  <variable name="that"/>
</action>
```

output: Outputs a WME by using the specified output-channel. If no output-channel is specified, PATE uses a default channel, i. e., the first output-channel that was registered.

```
<action type="output" channel="channelName" name="variableName"/>
```

cleanUp: Removes all WMEs from the working memory and the goal-stack. This action can be used to remove any remaining WMEs after a computation is finished.

```
<action type="cleanUp"/>
```

function: This action can be used to call a plug-in just like in the condition part (see section 7.3.1).

Realizing Time-outs

Sometimes the configuration of the working memory remains the same after a rule fired, for example, if the system has to wait for some additional input event while another object is on top of the goal-stack. Hence, the rules that just fired would be applicable in every cycle until the external event occurs, and the configuration of the working memory changes (or until their weighting factor is 0). To this end, PATE provides a feature that allows the elements on the goal-stack to expire after a specified amount of time. Before it is pushed onto the goal-stack, the expiring WME must have a slot `_timeout` added that contains the value (as string) of the expiration time in milliseconds. When expiring, an element is erased from the goal-stack and not stored in the working memory.

7.4 Applying Production Rules

In this section, we will discuss the general rule application mechanism of PATE. First, we will sketch how the conflict set of applicable rules is computed (see section 7.4.1), and how the firing rule is determined (see section 7.4.2). Then, we will outline how the firing rule can change the configuration of the goal-stack and the working memory (see section 7.4.3), and discuss how external events are added to the working memory (see section 7.4.4).

7.4.1 Computing the Conflict Set

PATE's central repeating fetch-execute cycle starts with the first WME inserted into the working memory or the first WME pushed to the goal-stack. The pseudo-code in algorithm 1 shows the processing steps that are executed during one cycle. In the first step (line 1), all rules that match the goal condition are computed and instantiated, i. e., the defined variables are bound. The second step (line 2) takes the remaining instantiated rules and checks if their remaining conditions can be fulfilled by the current configuration of the working memory. Each rule that passes this test is added to the final conflict set.

When all rules have been tested, the fetch-execute cycle continues with the evaluation of the applicable rules: if the conflict set is not empty (line 3), then the set of instantiated rules is sorted (line 4) according to their *rating value* (see the next section). The highest ranking rule is chosen to fire (line 5), which means its actions are applied (line 6). Furthermore, activation is increased for WMEs used by the rule, and decreased for all others (line 7). Finally, the dynamic weighting of the firing rule is decreased (line 8).

Algorithm 1 Fetch-Execute Cycle

```
1: goalConflictSet = computeConflictSet(goal-stack);
2: conflictSet = computeConflictSet(workingMemory, goalConflictSet);
3: if conflictSet not empty then
4:   sort(conflictSet)
5:   firingRule = #1(conflictSet);
6:   applyActions(firingRule);
7:   updateActivation();
8:   decreaseWeighting(firingRule);
9: end if
```

7.4.2 Selecting the Firing Rule

The rating of the instantiated rules of the conflict set (also called *probability of success*) is the most crucial part of a production rule system. Since examining every possible way that rules could fire would bind a lot of resources, it is important to identify the most promising rule in the conflict set for every fetch-execute cycle. In order to find the most promising rule, PATE utilizes a rating mechanism that rates every rule instance in the conflict set according to certain quality measures. The rule with the highest score will be chosen to fire.

The rating of a rule is based on a number of parameters. All actions of a rule are assigned a specific rating, defined in the configuration file of PATE. The sum of these ratings is normalized by dividing it by the maximum number of actions of all rules in the conflict set. To provide a mechanism preventing the same rule from firing every cycle, every rule has a

weighting ω_R , which decreases if the rule fires and increases if it does not. Nonetheless, it never exceeds a certain value which can be given by the weighting tag in the rule definition or by the default value 1.0. As rule evaluation is not an exact calculation, it is possible to modify the influence of some of the evaluation parameters. The ratings ω for the individual conditions and actions can be defined in PATE's configuration file. The rating of an instantiated rule is computed as follows:

Definition 8 Rule Rating

Let

R be a rule

ω_R be the current weighting of R

ω_W be the average activation of the involved WMEs

ω_C be the average weighting of the conditions of R multiplied by the unification or overlay score

ω_A be the average weighting of the actions of R

Then, *rule rating* ρ is defined as

$$\rho = \omega_R * (\omega_W + \omega_C + \omega_A)$$

The rating of a rule instance comprises the weighted activation of the WMEs (ω_W) used to match the conditions C , the weighting of the conditions (ω_C) the rating of the actions the rule is meant to perform (ω_A) and the rule's current weighting (ω_R).

7.4.3 Executing the Actions of the Firing Rule

The execution of the actions of the firing rule takes place in the same order as the actions have been declared and take effect immediately. This means that the configuration of the working memory is immediately changed after an action is applied. After the rule execution and just before the fetch-execute cycles are finished, a final post-processing step takes place where the weightings of the rules and the activation of the WMEs are adjusted. On one hand this means that the activation of all WMEs that were involved in the application of the firing rule is increased, and that for all the remaining WMEs the activation is decreased. On the other hand, the weighting of the firing rule is decreased by a factor specified in the configuration of PATE while the weighting of the remaining rules is increased up to the level of their default weighting.

7.4.4 Updating the Working Memory through External Events

A key feature of PATE is that it accepts new input from external applications at any time. This means it is always possible to add new information to the working memory so that it will be almost immediately available for rule application. However, to ensure a consistent state of the system, all incoming external events will be buffered and added to the working

memory at the end of the fetch-execute cycle. Technically, this is realized by means of two concurrent threads. The first thread called *PATEConnector* realizes the communication with the external components and provides two methods: *addToWorkingMemory(WME)* and *addToGoalStack(WME)*. Both methods register the specified WMEs for adding to the working memory via a scheduler that is provided by the second thread called *WorkingMemory*. This thread realizes the actual fetch-execute cycle and integrates all received external WMEs into the working memory at the end of the cycle.

If there is nothing to do for the fetch-execute cycle (i. e., the goal-stack is empty), this queuing of incoming events causes almost no delay (below 1 msec. on current standard PCs). However, if the goal-stack comprises at least one element, the actual adding of the WMEs is delayed until the fetch-execute cycle enters the last processing step.

7.5 Supporting the Process of Rule Development

7.5.1 The Graphical User Interface

With increasing complexity of the task and the number of rules, the process of writing rules and assessing their outcomes can become more susceptible to errors. To help maintain the system and to assist with writing the necessary rules, PATE has a built-in graphical user interface (GUI) for monitoring internal processing as well as providing features for altering the system state.

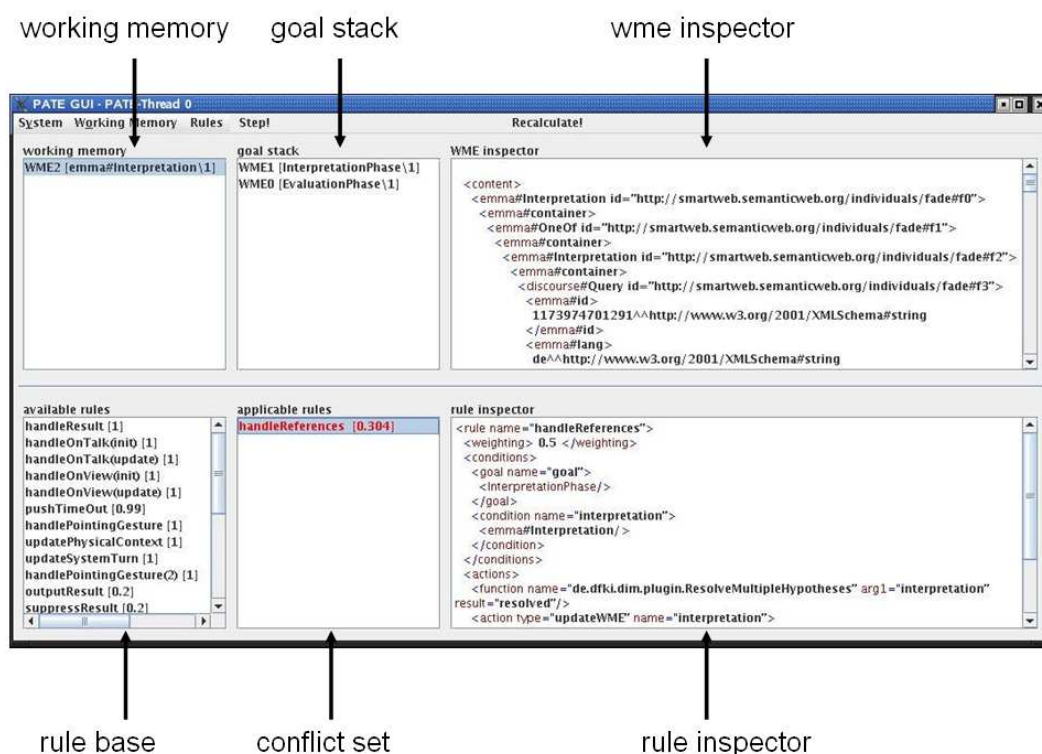


Figure 7.3: The graphical user interface of PATE.

The main view of the GUI (see figure 7.3) shows the state of the production rule system after calculating the firing rule but before applying the actions. In its upper part there are two lists, the working memory contents and the goal-stack, and a syntax highlighted XML viewer labeled WME inspector. The lists show the WME number, basic type and activation score. When one of the entries in the lists is clicked, the WME inspector shows the whole structure of the WME in an XML representation. Similarly, the lower part of the main view contains the lists of available rules and applicable rules, i. e., the calculated conflict set. Every rule is shown with its name and current weighting for the available rules, or its probability of success for the applicable rules. The current firing rule, shown in red, may be changed by selecting one of the other applicable rules. When one of the entries in a list is selected, the XML representation of the rule is shown, syntax highlighted, in the rule inspector.

The System Menu

The system menu of PATE contains basic functions for handling the system and the GUI.

System

Interactive Mode: Toggles interactive mode on or off. If interactive mode is on, the GUI shows the current system state while allowing it to be altered with the Working Memory menu (see section 7.5.1). Pushing the *Step!* button in the menu bar triggers exactly one fetch execution cycle. Disabling interactive mode causes the system to run without interruption until interactive mode is turned on again.

Load Configuration: Reads a new configuration file.

Set Trace Level: Opens a window where the new trace-level can be entered and sets PATE's trace-level to the specified value.

Update: Refreshes the information shown in all lists on the GUI.

Disable GUI: Hides the GUI, while the production rule system continues processing.

Exit: Shuts down PATE.

Working Memory

If the interactive mode is enabled, the Working Memory menu is accessible with the following functions:

Restart: Resets the production rule system. All contents of the working memory, goal-stack and the conflict set are deleted.

Add Goal: Pushes a new goal onto the goal-stack. Invoking this function opens a new text window where the eTFS structure of the new goal can be provided.

Add WME: Adds a new WME to the working memory. Invoking this function opens a new text window where the eTFS structure of the new WME can be provided.

Save State File: Saves all WMEs from the working memory and the goal-stack to an XML file.

Load State File: Adds all WMEs as defined in the XML text file created by the *Save State File* option to the working memory and goal-stack.

Delete WME/Pop Goal: Deletes the currently selected WME (as shown in the WME inspector) from working memory or from the goal-stack. Only the topmost element on the goal-stack can be deleted.

Rule

If the interactive mode is enabled, the Rule menu is accessible with the following functions:

Add Rule: Adds one or more rules to the production rule system. Invoking this function opens a text window where the new rule may be entered using the same syntax as for rule files as described in section 7.3. If there already is a rule with the same name, it is overwritten.

Edit Rule: Allows one to edit the currently selected rule of the list of available rules.

Save Rule File: Saves all available rules in an XML file.

Load Rule File: Loads a rule file. Adds the rules contained in the specified file to the production rule system.

The Step! button: Triggers the execution of a single processing step if the system is in interactive mode. If pressed, the *step!* button causes PATE to execute the actions of the selected applicable rule (if there is a rule in the list of applicable rules) and then to calculate the conflict set based on the new configuration. This feature is only available if the interactive mode is activated.

The Recalculate! button: Recalculates the conflict set without executing the currently selected applicable rule. This button can be used if the rule base has been updated and should be applied without any modification of the current configuration of the working memory. This button is only available in the interactive mode.

The Interactive Mode

If activated, the interactive mode switches PATE from automatic rule execution into a step-by-step mode. This mode enables the rule developer to monitor each single processing step and to intervene at certain points (i. e., by updating the rule base or selecting another rule to fire). If the interactive mode is activated, the normal fetch-execute cycle is paused just when the conflict set has been computed. At this point, the system halts until the user presses the *Step!* or *Recalculate!* button. The applicable rules (i. e., the conflict set) are displayed in the lower middle view of the GUI. The rule depicted in red is the rule that would fire if the system was not in the interactive mode.

Online Manipulation of the Conflict Set Normally, the rule with the highest rating is selected to fire. However, during the process of developing the rule base it happens that another rule should actually fire. For instance, if the weighting factor of the rule that is supposed to fire is too low, it might appear at a lower position in the conflict set. For the development of PATE rules it proved very useful to be able to intervene in the process of selecting the firing rule since the actual execution can be continued while the necessary updates can be made at a later point. The actual selection of the firing rule is straightforward. The developer just has to select the desired rule and to press the *step!* button.

The action-part of that rule is executed, and a new conflict set is computed based on the configuration of the working memory and the goal-stack. This feature, however, does not alter the rule base at all so that the rule developer has to manually change the respective rule if it is always supposed to fire given the current configuration of PATE (see next paragraph).

Online Manipulation of the Rule Base Another key feature of PATE is the possibility to update the rule base while the system is running. PATE provides two means to do this: (i) by adding new rules to the rule base and (ii) by editing existing rules. Moreover, because of the *Recalculate!* button it is possible to manipulate the rule, to generate a new conflict set, and then to continue at the same point as before the update.

The Why-Not Feature The *Why-Not* feature enables the rule developer to understand why a particular rule is not member of the conflict set. In order to access this feature, the developer just needs to select a rule that is not in the conflict set. After a short delay, a tool-tip appears and shows the name of the condition that did not match (or in case of a not-condition that did match).

7.6 Data Representation in PATE

The development of PATE's internal data representation API was guided by two conflicting aims: (i) to build an API that is as efficient as possible and (ii) to develop a representation format that is both expressive and flexible, and provides all required methods for manipulating and accessing the data.

Another important aspect of PATE's knowledge representation API is its re-usability. The API is designed in a way so that it can be easily employed for a different component or even a completely different system. But before PATE's knowledge representation is described in detail, we will discuss the requirement analysis that guided its development.

7.6.1 Requirement Analysis

From our experience with the development of several monomodal and multimodal dialogue systems, we compiled a set of requirements that has guided the development of PATE's knowledge representation framework from the very beginning. The following list shows the most important requirements:

Usability The overall goal is to develop a generic tool for developing modules for multimodal dialogue systems that is easy to use for rapid prototyping. Thus, the process of rule development must be integrated with extended testing capabilities.

Format The representation format has to be both easy to handle for human developers and to be efficient with regard to processing time. In addition, the representation format of PATE should support efficient implementations of operations like: *unification*, *restricted unification*, *overlay*.

Support of various input formats To facilitate the integration of PATE into a new host system or framework, it should support as many input and output formats as possible. The only restriction is that the format must somehow be related to the classic frame-based representation and support some kind of a type-hierarchy.

Type system The type system of PATE should support *multiple inheritance* and provide means to compute the sub-type and supertype relations as well as the least upper bound (see section 5.2.4). Furthermore, the type system should also support cardinality constraints.

7.6.2 Extended Typed Feature Structures

The data representation framework of PATE is called eTFS which stands for *Extended Typed Feature Structures*. This framework unifies features of RDF/RDFS, and typed feature structures and comes with a well-defined API.

The way we use eTFS to encode our internal data is similar to that of many ontology formats. An `<object>` tag denotes a complex object of a certain type (e. g., see the object of type `Song` in the eTFS example below). Complex objects in turn can be structured through the `<slot>` tag which defines named sub-features of a complex object. Atomic values are represented by the `<value>` tag. An object describing a song looks in the eTFS notation as follows:

```
<object type="Song">
  <slot name="artist">
    <object type="Artist">
      <slot name="firstName"> </slot>
      <slot name="lastName"> </slot>
    </object>
  </slot>
  <slot name="title"> One </slot>
  <slot name="length">
    <object type="TrackLength">
      <slot name="minutes"> 3 </slot>
      <slot name="sec"> 44 </slot>
    </object>
  </slot>
</object>
```

7.6.3 The Type System

The type system of the eTFS API realizes the type-hierarchy and provides methods for type comparison. Since most of the operations on eTFS instances require access to type information, the type system is a key aspect of the eTFS implementation. To this end, the type system is organized in an efficient way with a clear focus on fast response times for the individual methods.

The type system is organized in a hierarchical way where each instance of the class *Type* comprises information about its name, supertypes and slots. In order to improve the performance of the type system, each *Type* already comprises the complete list of its supertypes and slots. This means that the list of supertypes is directly available and there is no need for traversing the type-hierarchy to answer supertype relations between two types.

The type system is initialized by means of a layered post-processing of the type-hierarchy as it is defined by an ontology. In the first run, all defined types are traversed in order to update their supertype list and inherited slots. Moreover, each type is stored in the type-hierarchy with a unique integer as key so that the retrieval of type definitions is accelerated. In a second run, the post-processing process adjusts slots that might be defined more than once. This happens if an inherited slot is refined with a more specific slot filler. In this case, the more specific slot restriction will be kept.

The key methods of the type system of the eTFS API are:

Type getTypeForName(String t): Returns the type definition (**Type**) for the type with the specified name.

Type leastUpperBound(Type t1, Type t2): Returns the least upper bound (LUB) of the two specified types.

boolean subType(Type t1, Type t2): Returns true if t1 is a subtype of t2.

boolean superType(Type t1, Type t2): Returns true if t1 is a supertype of t2.

7.6.4 Multiple Slot-Fillers

With *multiple slot-fillers* we refer to a situation where an eTFS instance has multiple occurrences of the same slot. E.g., an instance of a class `Book` that has multiple occurrences of a slot named “chapter” with different slot-fillers (i.e., the individual book chapters). As mentioned by (Romanelli et al., 2005), operations like unification or overlay applied to standard TFSs show unpredictable results with respect to the treatment of multiple slot-fillers.

To this end, the eTFS type system incorporates the instance of cardinality constraints (see section 5.2.3) into the slot definitions for types. This means that every slot comprises, besides the type restrictions for slot-fillers, also information about the minimum and maximum cardinality of the slot.

7.6.5 Unique Identifiers

Unlike classic implementations of TFSs, our implementation also supports the denotation of instances with unique identifiers. The mechanism can be compared to the co-reference mechanism of classic TFSs, where two or more slot-fillers that appear on various levels of a TFS are linked together so that the content of the co-referenced objects will be the same. The unique identifiers used in eTFS, however, permit reference beyond the scope of individual documents. Consider, for example, the eTFS in figure 7.4 where the content of the first appearance of the slot *has_artist* is filled with an instance of the ID “artist42”. This instance is then again used as a slot-filler in the second appearance of the slot *has_artist* but could also be referenced in a different document.

7.6.6 Activation

Each eTFS object has an activation value defining its accessibility (see Anderson and Lebiere (1998)). This activation value has a range between 0 and 1 where 0 means that the object is not accessible anymore and 1 reflects the highest accessibility. The eTFS API provides methods to access and manipulate this activation value.

7.6.7 Supported Operations on eTFS Objects

The API of the eTFS framework supports a wide variety of operations to access, manipulate or maintain instances of eTFS objects. In the following we will briefly discuss a selected number of these operations.

```

<object type="Album">
  <slot name="has_track">
    <object type="Track">
      <slot name="has_song">
        <object type="Song">
          <slot name="has_artist">
            <object type="Artist" id="artist42">
              <slot name="has_firstName"> <value type="string"> Herbert </value> </slot>
              <slot name="has_lastName"> <value type="string"> Groenemeyer </value> </slot>
            </object>
            ...
          </slot>
        </object>
      </slot>
    </object>
  </slot>
  ...
  <slot name="has_track">
    <object type="Track">
      <slot name="has_song">
        <object type="Song">
          <slot name="has_artist"> <object type="Artist" id="artist42"/> </slot>
        </object>
      </slot>
    </object>
  </slot>
</object>

```

Figure 7.4: eTFS structure exemplifying the duplicate usage of the same identifier “artist42”.

Pattern Matching and Merging

The eTFS API provides three operations for pattern matching and merging of two eTFS objects: (i) restricted unification (ii) unification and (iii) overlay. Since these operations are central for the condition testing, the implementation of these operations are especially bound to be efficient and fast.

Restricted Unification A special case of unification is *restricted unification* which is non-commutative in contrast to standard unification (i. e., $rUnify(a, b) \neq rUnify(b, a)$). The idea behind this restricted unification is that the first argument specifies an information pattern that at least must be fulfilled by the second argument. Besides this, restricted unification works just like normal unification. As discussed in section 7.3.1, this operation is used for the condition matching part of PATE.

Unification The general functionality of the unification operation is discussed in section 5.2.4, so here we will only outline the basic algorithm of our implementation of unification. Because of the cycles that can be created using the enhanced ID mechanism, eTFS can no longer be considered directed acyclic graphs (DAGs). This has consequences for the implementation of the unification operation as the algorithm must be able to identify infinite loops and to take appropriate steps.

We also focused on the optimization of the unification algorithm in order to improve the performance with respect to processing time and memory consumption. The copying of the involved structures has been identified as a key issue. Tomabechi, for example, discusses an

approach which focuses on the idea of reducing the actual copying of structures by copying only those features that actually changed (Tomabechi, 1992). Since memory consumption is not that problematic nowadays, our implementation is centered around the idea of delaying the actual copying as long as possible so that as few features as possible have been copied when unification fails.

Scoring: Usually, unification does not provide any means for assessing the difference or similarity between the two TFSs involved. However, in some situations it is helpful to be able to assess how much information was added during the application of the unification operation. Thus, we realized a scoring function for the unification operation that computes how much discriminative information in terms of features was added from the second argument to the first argument of unification. Moreover, for some applications it is useful to assess the difference in terms of informational distance of the two arguments. To this end, unification also provides a measure of the distance of the types of the arguments.

Overlay The implementation of overlay in the eTFS framework differs from the previous implementation for the SMARTKOM system (see Pflieger (2002)) in three ways: (i) cardinality information (if available) is used for processing multiple slot-fillers, (ii) it supports multiple inheritance type-hierarchies⁴ and (iii) it introduces a mechanism to avoid overlaying instances that would lead to inconsistent instances (see section 5.2.4). While the first two aspects are handled in the algorithm, the third aspect requires an extension of the involved ontology. In order to be able to define which instances should not be overlayed, we employ a class *UnOverlayable* that marks all types that are sub-classes of that class as excluded from being overlayed.

Viewed on an abstract level, overlay operates as follows: First, the cover and the background are tested as to whether they are instance of the type *UnOverlayable*. If this is the case, they will not be overlayed and the cover will be returned. Otherwise, overlay will loop over the set slots of the cover and the background and recursively overlay the slot-fillers. The implementation of overlay in eTFS supports both scoring mechanisms that are discussed in section 5.2.4.

Retrieving Embedded Objects in Complex Structures

Searching for particular objects that are embedded in large eTFS structures can be a costly and time-consuming task. The eTFS API provides three different methods for looking up substructures of complex eTFS:

eTFS[] findObjectOfType(Type type): Returns all objects of the specified type that are embedded in an eTFS object.

eTFS[] findObjectOfTypeOrSubtype(Type type): Return all objects of the specified type or a subtype thereof that are embedded in an eTFS object.

⁴With the restriction, however, that it produces only one result. This stands in contrast to the findings of Alexandersson and Becker (2006) who showed that in case of very special inheritance constructions, some specific pairs of types can have more than a single least upper bound. Alexandersson and Becker have argued that in such a case overlay would need to return more than one result. But since these inheritance constructions never occurred in our domains, eTFS demands for type hierarchies where each pair of types has not more than a single least upper bound.

eTFS[] findObjectsThatUnify(eTFS tfs): Returns all embedded objects of an eTFS object that unify with the specified pattern.

Accessing eTFS Objects by Feature Paths

For the accessing and the manipulation of embedded substructures of eTFS objects, we have developed a simple and lightweighted *feature path* representation based on XPATH⁵ that is used by a method called `getXPath` for accessing substructures of eTFS objects. A feature path defines a sequence of slots of where to find a particular object. In our implementation, the individual slots of a features path are separated by means of two colons: “::”. Given a feature path like “has_addressee::has_name” the algorithm traverses the specified eTFS object starting with the root node. For each slot of the feature paths, it determines whether the eTFS object contains this slot and if the slot is filled it goes into recursion. This continues until the complete feature path has been traversed or until the algorithm encounters an empty or undefined slot.

Moreover, the algorithm is also able to deal with multiple defined slots. Here, it is possible to directly access a particular item of the set of slots by providing the number of the slot using squared brackets (e.g., “has_addressee[1]::has_name”) or to go with all possibilities which means the algorithm would return all suitable objects it encounters during the traversal.

Manipulation of eTFS Objects

For the manipulation of substructures of eTFS objects, the eTFS API provides a method called `setXPath` for replacing embedded objects. This method basically operates like the `getXPath` method as it traverses a given eTFS object using the specified feature path. However, instead of returning the target object, `setXPath` replaces it with the specified eTFS object.

7.7 Using PATE as a Generic Component for Multimodal Dialogue Systems

Besides the application within FADE, the PATE system has been and is successfully used in a number of projects (e.g., COMIC, TALK, SMARTWEB, VIRTUALHUMAN, I2HOME) and fulfills a variety of dialogue-related tasks in these systems. In the following section we will briefly discuss the SAMMIE system which is the first system whose core components have been completely realized with PATE.

7.7.1 SAMMIE—An In-Car Multimodal Dialogue System

The SAMMIE system (Becker et al., 2006a,b) was developed within the TALK project⁶ and is an example for an advanced multimodal dialogue interface that provides state-of-the-art interaction capabilities of a research prototype combined with a uniform and maintainable knowledge representation needed for commercial applications⁷.

⁵See <http://www.w3.org/TR/xpath>.

⁶See <http://www.talk-project.org/>.

⁷Most of this material can be found in Pflieger and Schehl (2006).

SAMMIE provides a multimodal interface to an in-car MP3 player through speech and haptic interaction using a BMW iDrive (see section 4.3.2). System output is realized by speech and a graphical display integrated into the dashboard of the car. An example of the system display is shown in the lower right part of figure 7.5.

The architecture of the SAMMIE system follows the classic approach of a pipelined architecture (Bunt et al., 2005; Maybury and Wahlster, 1998). Figure 7.5 illustrates the modules and their interaction: Modality-specific recognizers and analyzers provide a semantic representation of the recognized input that is sent to the Multimodal Fusion module. This component in turn interprets them in the context of the other modalities and the current dialogue context (maintained by the Discourse Module). The Dialog Manager implements the collaborative problem solving (CPS) approach (Blaylock, 2005; Allen et al., 2002). It decides the next system move based on its task model, the current context, and also on the result from calls to the MP3 database. The Multimodal Fusion module then generates an appropriate system response by planning the actual content, distributing it over the available modalities, and by finally coordinating and synchronizing the output. Finally, modality-specific output modules generate the spoken output and an update on the graphical display. All modules interact with the Discourse Module in which the available context information is stored. The key to the SAMMIE system is that all of the core tasks, namely discourse modeling, interpretation/fusion, dialogue management and linguistic and presentation planning, are realized using PATE.

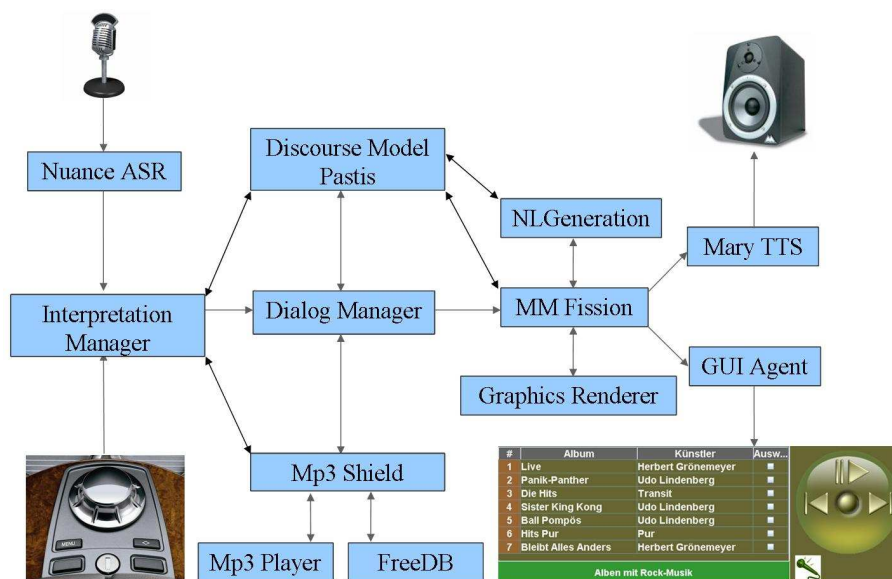


Figure 7.5: System architecture of the SAMMIE system.

7.8 Conclusion

With the PATE system we have introduced a production rule system that is capable of dealing with the reactive aspects and controlling the processing flow of FADE. However, it also serves

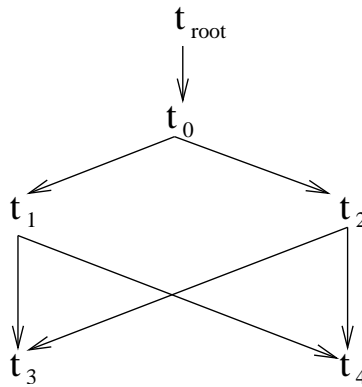


Figure 7.6: Sample type-hierarchy where overlay would return more than one result (adapted from Alexandersson and Becker (2003)). Since the types t_3 and t_4 have no unique least upper bound (LUB) but two minimal upper bounds (MUBs), the result of applying overlay might consist of two instances, one for each MUB.

as a generic tool for the development of components for ontology-based dialogue systems. A key feature of PATE is that it directly operates on ontological instances and that it provides easy means to access and manipulate these instances. The eTFS API is a lightweight framework for the representation of ontological data that supports a wide variety of data manipulation and pattern matching operations (including unification, restricted unification and overlay). It extends the classic TFS approach by introducing unique identifiers. By means of these identifiers, it is possible to create unique entities reflecting instances of ontological objects. A limitation of our current implementation of unification and overlay might be the fact that it does not support multiple results. As Alexandersson and Becker (2006) discuss, there are situations where a single application of overlay might return more than one result. These situations are in particular triggered by means of specific inheritance relations within the type-hierarchy. Figure 7.6 shows an example of such a cross-inheritance relationship. Following Alexandersson and Becker (2003, 2006), the types t_3 and t_4 in figure 7.6 have no unique LUB but two Minimal Upper Bounds (MUBs). If applied to instances of these two types overlay might return two results, one for each MUB.

Another key feature of PATE is the plug-in mechanism that enables application developers to perform computations that cannot be expressed using production rules. Due to the built-in interactive mode which supports online rule development, PATE facilitates the rapid development of applications. However, there are three features usually included in production rule systems that are not included in PATE: rule compilation, meta-rules and specificity as a rule selection criterion. All three features are related to the process of selecting the firing rule out of the conflict set. In PATE, this process is influenced only by the weighting the rules of the conflict set and selecting the rule with the highest score to fire.

Chapter 8

An Integrated Fusion and Discourse Engine

While the previous chapters discuss the notion of a comprehensive context model and introduce with PATE a framework for developing components for multimodal dialogue systems, this chapter will now show how these parts are incorporated into the FADE system. FADE is a generic and reusable engine for multimodal fusion and discourse processing. We will start with a brief consideration of the foundations of FADE, and then describe the basic architecture of this component in section 8.3. The following three sections discuss the functional and technical aspects of the three main components of FADE: (i) the perception module (section 8.4), (ii) the immediate conversational context (section 8.5) and (iii) the discourse context (section 8.6).

8.1 Introduction

This section gives an overview of the basic functionality of FADE and provides some examples that illustrate how FADE can be used to handle the aforementioned multimodal phenomena. At this point, we will only give a high-level overview of FADE's underlying architecture; for more details see the next section. The central processing and reasoning component of FADE is the production rule system PATE already introduced in Chapter 7 (see figure 8.1).

As discussed in Chapter 7, PATE introduces several extensions to classic production rule systems, in particular the activation-based processing of knowledge chunks. The main purpose of PATE in the fusion and discourse engine of our dialogue system is to control the collection and synchronization of incoming monomodal events (i. e., processing the *interactional* contributions) and to manage the resolution of referring and elliptical expressions via the discourse modeler (i. e., processing the *propositional* contributions). Moreover, PATE can be used to deal with system specific input that cannot be handled by the built-in processing logic of FADE.

As will be shown later, using a production rule system like PATE for the synchronization and first interpretation of monomodal input events permits the implementation of reactive system behavior. In many multimodal dialogue systems, the synchronization of incoming monomodal events is realized by means of rather simple time-outs, for example, “*wait for x secs., collect whatever you receive, integrate everything.*” This fusion strategy, however, has the drawback that the system has to wait for the time-out even if the user clicks on a link

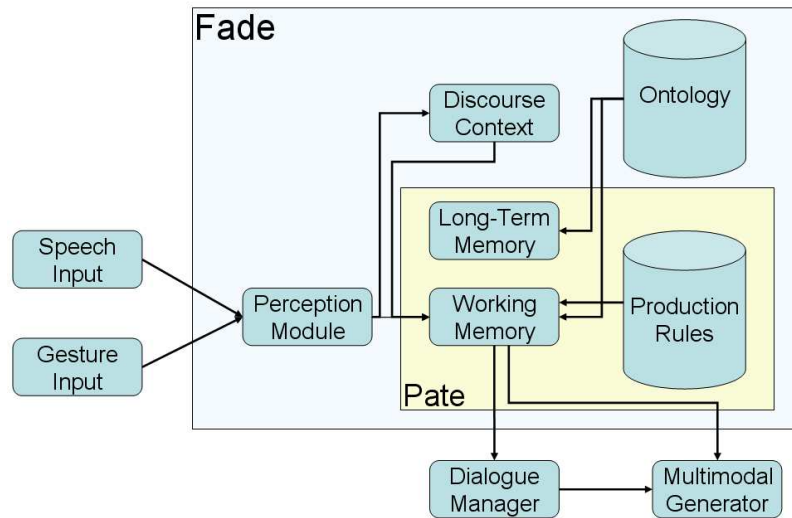


Figure 8.1: Basic architecture of the integrated Fusion And Discourse Engine (FADE).

which could immediately be processed. Using PATE, it is possible to state specialized rules that directly trigger appropriate reactions, in case of unambiguous input events that can be mapped to a distinct interpretation. Using FADE it is even possible to combine these two approaches, i. e., to employ a time-out only if necessary.

As discussed in Chapter 6, our context model is based on the idea of a separated processing of interactional and propositional contributions. To this end, we differentiate between two main contextual areas: (i) the immediate conversational context and (ii) the discourse context. These two context areas are realized by PATE and DiM respectively.

Moreover, we also distinguish between a working memory (WM) and a long-term memory (LTM) as the central repositories for knowledge chunks. The working memory comprises all the activated entities, i. e., those entities that have been referenced during the discourse. The LTM in turn is the central knowledge repository of FADE and it comprises all predefined knowledge the system is aware of. The entries of the LTM are also assigned an activation value that spreads between associated entities. Between the WM and the LTM, a transfer of knowledge chunks can take place if the activation of an entity that is currently located in the LTM exceeds a certain threshold. Then this entity appears in the WM. As soon as the activation of that entity falls below this threshold it disappears from the WM and can only be accessed in the LTM.

In this chapter, we will give an overview of the functional and technical aspects of FADE. But first a brief overview of a meta-ontology developed for modeling domain-independent information required by FADE will be given.

8.2 An Ontological Model of Conversational Dialogue

In this section we describe FADE's meta-ontology which models all concepts that are required for FADE. These concepts have to be integrated or inherited into the domain specific ontology of the target system. For the development of this meta-ontology we used the Protégé 3.1

ontology editor that is provided by Stanford University¹. Protégé is a free, open source ontology editor and knowledge-base framework and supports several output formats such as RDFS and OWL.

The key idea of this meta-ontology is that it provides a fixed set of ontological classes that can be extended by a domain specific ontology. The knowledge source of FADE that defines the generic aspects of the processing strategies can make use of these meta-classes and thus can be reused for different applications. The total number of required meta-concepts is rather small compared to the typical size of ontologies for multimodal dialogue systems.

In total, FADE's meta-ontology comprises four main concept branches plus a set of utility concepts. The following subsections will briefly introduce these concepts. We will start with the representation of verbal and nonverbal actions and then briefly discuss the modeling of the immediate conversational context (see section 6.2). Then, we will discuss the modeling of the physical environment and that of referring expressions.

8.2.1 Modeling Verbal and Nonverbal Actions

Typically, an ontology for a multimodal dialogue system such as the SMARTWEB ontology or the VIRTUALHUMAN ontology comprises concepts for modeling the communicative intention of user utterances. As discussed in section 2.2.4, dialogue acts are a common way to express this kind of information. To this end, we based our taxonomy for verbal and nonverbal actions on the notion of *Acts* as a general super-type for verbal actions (i. e., dialogue acts) and nonverbal actions. Figure 8.2 gives an overview of the top-level of this sub-taxonomy.

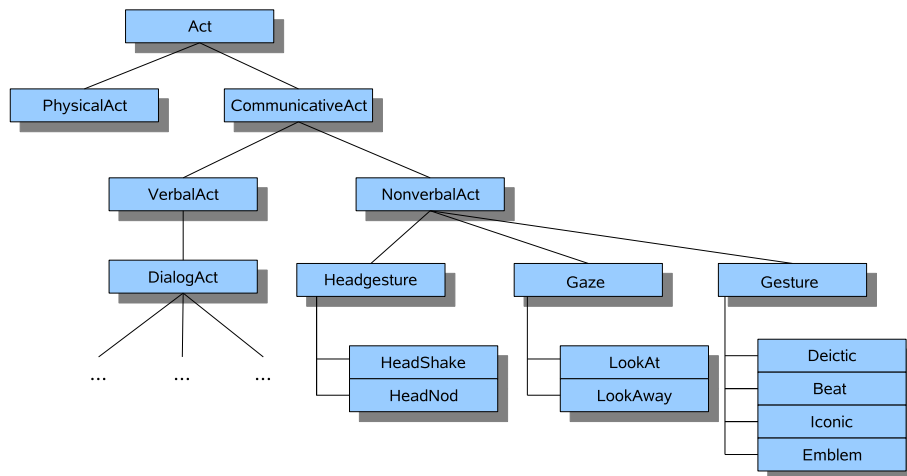


Figure 8.2: The upper-level of FADE's internal taxonomy for *Acts*.

Each *Act* comprises at least the following slots:

has_initiator—describing the participant of the interaction that conducted this *Act*.

has_content—describing the actual semantic content of that *Act*.

¹See <http://protege.stanford.edu/>.

has_beginTime—describing the point in time when the realization of that **Act** started.

has_endTime—describing the point in time when the realization of that **Act** was finished.

At the level of the class **CommunicativeAct**, a slot called **has_addressee** (allowed slot-fillers are instance of the class **Participant**) is introduced. This slot can be used to specify the addressees of a communicative action.

8.2.2 Modeling the Conversational Status

As discussed in section 6.2.2, the purpose of the conversational status is to model all aspects of an ongoing interaction that affect the interpretation of perceived monomodal contributions of the participants. This enables FADE to handle an extended set of phenomena as they arise in multimodal multiparty interactions. The ontological model of the conversational status incorporates all aspects of the conversational status that is outlined in section 6.2.2.

At the top-level of the meta-ontology, the conversational context is realized by means of the class **ConversationalStatus**. This class comprises the following slots:

has_turnDuration—describing the duration of the current turn in milliseconds.

has_speaker—describing the current speaker (must be an instance of **Participant**).

has_addressees—describing the current participants (if already inferable; must be instances of the class **Participant**).

has_sideParticipants—describing the current side participants (must be instances of the class **Participant**).

has_overhearers—describing the current overhearers (must be instances of the class **Participant**).

has_previousSpeaker—describing the previous speaker.

has_previousAddressee—describing the addressees of the previous turn.

has_activeFrameOfReference—describing the currently active frame of reference (allowed fillers are *intrinsic*, *relative*, *absolute*).

has_activeTemporalReferencePoint—describing the currently active temporal reference point (allowed slot fillers are instances of the class **TimePoint**).

8.2.3 Modeling the Physical Environment

As discussed in Chapter 3, exophoric references to the physical environment are a common instrument in situated dialogues. To this end, the context model of FADE comprises an elaborated representation of the physical environment that represents the basic spatial relations between objects visible in the scene (see section 6.2.3). This representation of the physical environment is incorporated into FADE's ontology.

At the top-level, the physical environment is modeled by the class **PhysicalEnvironment**. This class has three different slots:

`has_PEType`—describing the type of the spatial arrangement, e. g., `DoubleColumnList` or `List`. (allowed slot fillers are all subclasses of the class `PEType`; see below for a list of supported types).

`has_orientation`—describing the orientation of the entire physical environment (allowed slot-fillers are: *northern, eastern, southern, western*).

`has_absolutePosition`—describing the individual objects located in the physical environment (allowed slot fillers are instances of the class `AbsolutePosition`).

The class `PhysicalEnvironment` itself is also a subclass of `AbsolutePosition` so that it is possible to nest closed physical environments within another physical environment (this feature can be used, for example, to model a house that consists of rooms). But note that the model described here is not intended to provide a comprehensive representation of all aspects of spatial relations between Physical Objects as it is tailored to the needs of the spatial reference algorithm which will be discussed in section 8.6.2. As already indicated, the class `AbsolutePosition` is used to encapsulate the individual objects in a physical environment. It comprises the following slots:

`has_orientation`—describing the orientation of the object (allowed slot-fillers are *northern, eastern, southern, western*).

`has_northernNeighbor`—describing the northern neighbor of an `AbsolutePosition` which must again be an `AbsolutePosition`.

`has_easternNeighbor`—describing the eastern neighbor of an `AbsolutePosition`.

`has_southernNeighbor`—describing the southern neighbor of an `AbsolutePosition`.

`has_westernNeighbor`—describing the western neighbor of an `AbsolutePosition`.

`has_ontologicalInstance`—holding the semantic description of the Physical Object associated with an `AbsolutePosition`.

`has_positionDescriptor`—holding the semantic description of the position. This is an optional slot that is only used for discretely labeled positions (e. g., the positions on a football field).

8.2.4 Modeling Referring Expressions

For the representation of referring expressions, FADE's meta-ontology makes use of a class called `RefProp`. At this point, FADE's meta-ontology itself incorporates concepts from a different ontology that is called the *lingInfo* ontology (see Buitelaar et al. (2006)) and that is being developed within the SMARTWEB project.

An instance of the class `RefProp` can be used to denote that a particular instance represents a referring expression. The actual usage of this mechanism is discussed in more detail in section 8.6.2, here we will only discuss its underlying ontological model. The class `RefProp` introduces the following slots²:

²The namespace prefix `lingInfo` denotes concepts that are incorporated from the *LingInfo* ontology.

has_type—describing the type of the referring expression (allowed slot-fillers are *indef*, *def*, *deictic*, *trueDeictic*).

has_differentiationCriterion—describing the differentiation criteria (see below) if available.

has_morphoSyntacticDecomposition—describing the linguistic properties of the surface realization (allowed slot-fillers are instances of the class `lingInfo:WordForm`; see below).

The class `lingInfo:WordForm`, as defined in the `LingInfo` ontology, comprises the following slots (besides others):

lingInfo:has_case—describing the case of the surface realization.

lingInfo:has_gender—describing the gender of the surface realization.

lingInfo:has_number—describing the number of the surface realization.

lingInfo:has_partOfSpeech—describing the assigned part-of-speech tag (if available).

The class `DifferentiationCriterion` is used to denote restrictions in order to distinguish the referent out of a set of possible referents (e. g., entries in a list).

has_DCValue—describing the actual differentiation criterion (e. g., “third” for the third element of a list).

has_order—describing the starting point from which the differentiation criterion should be applied.

An example of a differentiation criterion is the reference to a list element such as “*the third song from the back*”. This results to the following instance of the class `DifferentiationCriterion` embedded in a `RefProp` instance:

$$\left[\begin{array}{l} \text{RefProp} \\ \text{has_differentiationCriterion:} \\ \\ \text{has_morphoSyntacticDecomposition:} \end{array} \left[\begin{array}{l} \text{DifferentiationCriterion} \\ \text{has_DCValue:} \quad 'third' \\ \text{has_order:} \quad 'reverse' \\ \text{lingInfo:WordForm} \\ \text{has_gender:} \quad 'neuter' \\ \text{has_number:} \quad 'singular' \end{array} \right] \right]$$

Besides this type of differentiation criterion, there is another type for representing spatial expressions like “*the guy left of Klose*”. These referring expressions can be represented by means of the class `SpatialExpression` which extends the class `DifferentiationCriterion`. The class `SpatialExpression` adds the following slots to the ones that are inherited from `DifferentiationCriterion`:

has_spatialExpression—describing the type of spatial relation between the relatum and the referenced object (allowed slot-fillers are instance of the class `SpatialExpression`, see the next section).

has_relatum—holding a semantic description of the relatum.

has_referenceObject—holding a semantic description of the referenced object.

8.2.5 Utility Classes

In the following we will briefly mention some of the utility classes that will be mentioned within this thesis.

The class `PType` currently has the following subclasses but can be easily extended with additional subclasses: `List`, `DoubleColumnList`, `GroupPicture`, `Map` and `FootballField`.

The class `SpatialExpression` is used to represent the spatial relationship that holds between two objects in the physical surroundings. Currently, it comprises the following subclasses: `Above`, `Below`, `Left`, `LowerLeft`, `LowerRight`, `Right`, `UpperLeft`, `UpperRight`.

Discourse participants can be modeled in various ways, and often top-level ontologies already comprise detailed classes for modeling agents, or humans that can be reused for modeling discourse participants. FADE's meta-ontology provides a class that represents discourse participants with all the information needed for the tasks at hand. This class is called `Participant` and comprises the following slots³:

has_name—describing the participant's first name and last name (but also nicknames).

has_sex—describing the participant's sex (*male*, *female*).

has_activeNonverbalBehavior—describing the currently active nonverbal behavior.

has_frameOfReference—describing the currently used frame of reference (*intrinsic*, *relative* or *absolute*).

has_position—describing the participant's current position in the scene.

has_emotionalState—describing the participant's emotional state (see, for example, Gebhard (2005)).

Note that there are two notions of the active frame of reference within the conversational status. A global one that is represented at the top-level of the conversational status and a local one that is represented at the individual participants. The global frame of reference describes the generally accepted frame of reference and the local one can be used to represent individual differences in case a specific participant uses a different frame of reference.

8.3 The Basic Architecture of FADE

The high-level functional architecture of FADE consists of two processing layers: (i) the production rule system PATE which is responsible for the reflexive interpretation of perceived monomodal events and (ii) a discourse modeler (called DiM) that is responsible for maintaining a coherent representation of the ongoing discourse and for the fusion of multimodal contributions and the resolution of referring and elliptical expressions.

The system needs additional interfaces and sources of information to obtain full functionality. Besides the common interfaces that can be found in standard architectures for multimodal dialogue systems as depicted in figure 4.2, FADE needs a set of additional interfaces to a number of other modules (e. g., to the (multimodal) generation component and

³Note that this model only covers the basic aspects necessary for the tasks at hand. However, there are more comprehensive user models put forward that are used to represent users in more detail. Consider, for example, the GUMO ontology (Heckmann, 2006).

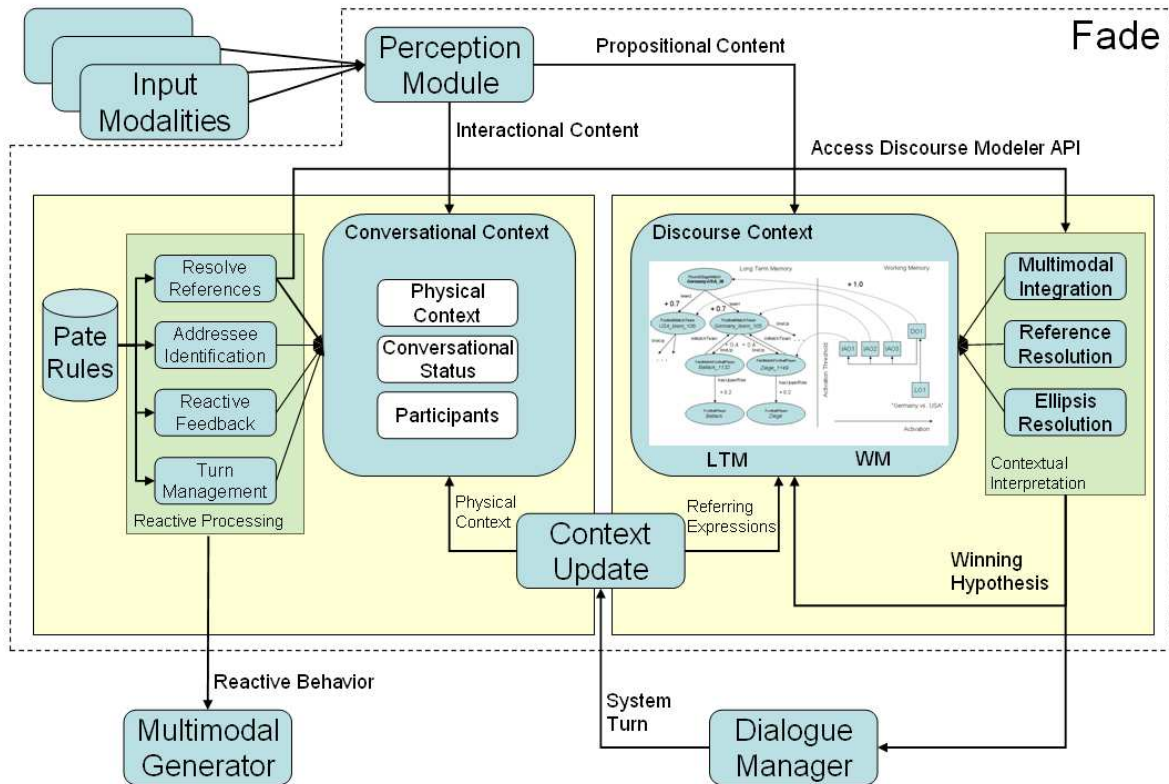


Figure 8.3: The functional architecture of FADE. The left part of the figure shows the conversational context that is accessed and updated by PATE rules and the right part shows the discourse context and its API for processing propositional contributions.

the recognizers). These interfaces enable FADE to immediately trigger turn-taking related signals and backchannel feedback as well as to react properly to changes in the surrounding world.

The key to our approach is that every recognition component produces incremental output and informs FADE as early as possible that some input has just been recognized, but only when a certain confidence level is reached. This in turn enables FADE to trigger appropriate reactions via the generation module depending on the current state of the discourse (i. e., if the floor was just available and the speech recognizer reported the beginning of an input event, a proper reaction would be to look at the participant who has just started to speak). The generation of appropriate backchannel feedback, however, is a little more complicated. This kind of feedback needs to reflect the current understanding of the perceived contributions. In short, the discourse modeler validates the incrementally analyzed contributions of the speaker with respect to the discourse context and signals possible misunderstandings to the local context. The local context in turn identifies potential gaps in the speaker's turn into which backchannel feedback can be inserted (see Pflieger and Löckelt (2005)).

This architecture poses high demands on the input and analysis components. For instance, an incremental input processing and real time operation is crucial for the processing and

generating of backchannel feedback. Any serious delay of reactive behavior in the virtual character would spoil the illusion of a human-like conversation.

8.4 The Perception Module

As depicted in figure 8.3, all input that is sent to FADE is routed via the *perception module*. The purpose of this perception module is to assign appropriate activation values to perceived objects or events and to route the received data to the appropriate context model.

If the received object stems from modality specific recognizers or analyzers, the perception module will use, if available, the confidence value as activation value⁴. The computation of the activation of objects that are part of the physical surroundings is somewhat more complex and will be described in the following subsection.

8.4.1 Determining the Activation of Verbal and Nonverbal Actions

For each incoming monomodal event, the perception module has to compute an appropriate activation value. Since these objects run through the recognition and analysis stages, they are typically assigned a confidence score reflecting the recognizer or analyzer confidence in its result. If available, these confidence scores are used by the perception module in order to initialize the activation value for that object.

8.4.2 Determining Visual Saliency

Determining visual saliency of the objects of the physical surroundings is a complex task. As discussed in section 4.4.3, a number of approaches with varying complexity and coverage have been put forward. The approach of Maass (1996), for example, models visual saliency based on specific characteristics of the Physical Objects like color, height and width of objects. However, determining visual saliency based on these criteria requires complex reasoning involving domain knowledge. In order to preserve the generic character of FADE, the perception module employs a simplified algorithm that determines the activation of Physical Objects without considering their physical qualities.

The activation value of all objects perceived in the visual environment is determined based on the visual saliency model presented in (Byron et al., 2005). This model is based on three parameters: **Uniqueness (U)**, **Recency (R)** and **Persistence (P)**. Even though the original algorithm is designed to model visual attention in a scenario where the visual field frequently changes (i. e., the agents move around in a virtual world), it is also useful to model the saliency for graphical displays where the location is relatively static. The following description is based on the one of (Byron et al., 2005, pp. 98), but it contains some modifications that we had to add for the algorithm to fit our purposes.

Uniqueness At the beginning of an interaction, all objects of the physical environment are initialized with the maximum value 1 for U (which means that all objects are equally unique). The actual algorithm for computing the uniqueness is based on a time window called the *uniqueness window* (T_u). The uniqueness value of an object at time point t is

⁴Note that the activation value must not exceed the interval between 0 and 1. Thus, FADE only accepts confidence values that are within this interval.

based on its previous uniqueness value at $t-1$ minus a quantity proportional to the frequency of its occurrence in the field of view:

Definition 9 Uniqueness (U) Let

i be the object

j be the current time

$n_{i,j}$ be the number of times object i was seen between $j - T_u$ and j .

k be a constant factor denoting the proportionality between the penalizing factor δ and the frequency of occurrence of i ⁵

$$U_{i,j} = U_{i,j-1} - \delta$$

$$\delta = k * \left(\frac{n_{i,j}}{T_u}\right)$$

Objects that appear and disappear frequently receive a large δ value when computed over a small T_u which means that its uniqueness value is low. As Byron et al. note, this would be “consistent with the phenomenon of object-based Inhibition-of-Return (IoR)” (Byron et al., 2005, p. 90).

Persistence The recency values of different objects should not only depend on the length of the time interval since the object was last seen but also on how long they were visible before disappearing. Persistence measures the exposure time of an object as the frequency of occurrence of an object within a time interval, called the *persistence window* (T_p):

Definition 10 Persistence (P) Let

$m_{i,j}$ be the number of times an object i was seen between $j - T_p$ and j

$$P_{i,j} = \frac{m_{i,j}}{T_p}$$

The persistence parameter is weighted by a factor c :

$$\sigma = c * P_{i,j}$$

⁵Note that all weighting factors can be configured in the configuration file of FADE (see section 9.1.2).

Recency If an object dropped out of the visual field, its probability to be referenced by a referring expression decays with time. As Byron et al. note, this is in-line with “well known decay of visual memory with increase in recall time” (Byron et al., 2005, p. 90). Byron et al. chose a zero-centered Gaussian to model the recency factor⁶:

Definition 11 Recency (R) Let

σ be the weighted persistence parameter

$t_{i,j}$ be the length of the time interval measured from j since object i was last seen

$$R_{i,j} = e^{-\left(\frac{t_{i,j}}{\sqrt{2}\sigma}\right)^2}, (t \geq 0)$$

Note that all objects currently visible have a maximum recency value of 1.

Activation of Visually Perceived Objects The actual activation or salience of a visually perceived object that is currently or that has been part of the physical context is based on the uniqueness and recency factors of the object:

Definition 12 Activation (A)

$$A_{i,j} = U_{i,j} * R_{i,j}$$

For each object of the physical environment, the perception module computes its individual activation based on this algorithm. This activation value is updated when a new turn of a participant (e.g., the user) starts or when the configuration of the physical environment changes.

To get a better understanding of how the activation of physical objects influences the interpretation process, consider the following situation⁷: The system displays a large list of MP3 songs on the screen. When the user starts scrolling down this list, the activation values of the individual songs change constantly which directly affects their accessibility. If the list contains two songs with the same name (i.e., “One”), a spoken command like “*Play One.*” would be ambiguous (i.e., it is not clear which of the two songs the user is referring to). However, using the algorithm for determining visual attention provides FADE with a weighted representation of the list based on salience. As a consequence, one of the two appearances of the song “One” has a higher activation compared to the other which enables FADE to resolve the exphoric reference (the actual reference resolution algorithm is discussed in section 8.6.2).

⁶As Byron et al. highlight, the zero-centered Gaussian leads to “a slow decay in R immediately after an object disappears, followed by a period of rapid decay that leads to an almost constant near-zero value” (Byron et al., 2005, p. 90).

⁷This example is taken from a new multimodal dialogue system called *AMADA* into which FADE is being integrated at the time of writing. This system provides a multimodal interface for controlling a MP3 player in an in-car scenario (see section 10.2).

8.4.3 Event Routing

For the routing of incoming events (i. e., the classification into interactional and propositional contributions), the perception module pushes the pre-processed objects onto PATE's goal-stack. The actual routing is then realized by means of PATE rules. This has the advantage that any changes in the representation format that would affect the classification concerning their interactional or propositional content is not represented in the code base but can easily be accounted for in PATE's rule base.

8.5 The Immediate Conversational Context

The processing logic of the conversational context is implemented by means of PATE rules. This gives a great flexibility for adapting the system to new tasks or new ontological models. However, this also means that the actual rules depend to some extent on the ontological structure and the specifics of the target system. To this end, we describe here only the high-level aspects of the individual functionalities that can be realized with the conversational context. The actual realizations of these rules will be discussed in Chapter 9. Furthermore, for most applications only a subset of these functionalities is required.

8.5.1 Representation and Maintenance of the Conversational State

The conversational state is represented by means of the ontological structures introduced in section 8.2. But the actual level of detail depends on the requirements of the dialogue system and the task at hand. It ranges from modeling only a small excerpt of the perceived actions of the user up to modeling all aspects of the conversational setting (including the conversational roles of the participants and the physical environment) plus the individual participants (including the system). For a task-oriented dyadic dialogue system it could, for example, be sufficient to model only the key actions of the user and to ignore information regarding the turn-taking process that goes beyond the detection of who is the current speaker.

Updating the Immediate Conversational Context

For the immediate conversational context it is vital that it is updated on a regular basis. This means every action or event should immediately be incorporated into the immediate conversational context. As mentioned, this has to be realized by means of PATE rules that are tailored to the special needs and data structures of the target system. In order to facilitate the adaptation process for the application developer, FADE already provides a set of design patterns and rule bodies that can be easily adapted to the new application.

Triggering Reactive Actions

Depending on the capabilities of the dialogue system, FADE can be used to trigger reactive actions in response to incoming events. Again this has to be realized by means of PATE rules that test for the existence of particular monomodal events in certain configurations of the immediate conversational context. In the following two subsections we will discuss how this can be done for interpreting and triggering turn-taking signals and backchannel feedback.

Triggering and Interpreting Turn-Taking Signals As discussed in section 3.3.2, the turn-taking system consists of a number of signals that are displayed by the speaker and the addressees. In some situations, these signals can be ambiguous and require contextual information in order to be interpreted in the right way.

For the VIRTUALHUMAN system we implemented the key aspects of the turn-taking model described in (Pfleger and Alexandersson, 2004) (see also section 3.3.2). Only those aspects that require information about the eye and hand movements of the users have not yet been implemented due to the lack of corresponding sensory information. However, given the availability of the respective sensory data, it would be straightforward to implement PATE rules that cover the complete turn-taking. In section 9.2.5, we will discuss how we implemented the turn-taking management for the VIRTUALHUMAN system.

Triggering and Interpreting Backchannel Feedback As discussed in section 3.3.4, backchannel feedback can be used to display the current understanding of a listener while the speaker is still speaking. This means, however, that the listener needs to be able to incrementally process the contributions of the speaker and to immediately display appropriate feedback. Thus, triggering backchannel feedback requires for real-time analysis of the perceived actions of the current speaker. Since the dialogue systems into which FADE has been integrated do not support incremental processing, we implemented prototype rules that are able to trigger general backchannel feedback like gazing behavior or head nods. These types of backchannel feedback can be displayed during a broader time window but provide no direct feedback with respect to the grounding process. Section 9.2.5 outlines how this type of backchannel feedback is realized for the VIRTUALHUMAN system.

The interpretation of backchannel feedback also requires for fast decisions and thus for a reactive processing architecture (see section 4.1.1). Even though it has not yet been used in an actual system, we developed a basic processing strategy for dealing with backchannel feedback. The overall idea is that speakers constantly monitor the feedback of the other participants so that they are able to react to negative feedback.

The interpretation of backchannel feedback is only active while the participant a particular FADE instance stands for is the current speaker. It is realized by means of PATE rules and needs access to the temporal organization of the generated utterance (i.e., the time-spans for realizing the individual concepts). The actual interpretation can be realized using the processing strategy as depicted in algorithm 2.

Algorithm 2 Interpretation of Backchannel Feedback

```

1: classify received feedback into: positive, neutral, negative
2: if type = positive or type = neutral then
3:   identify corresponding concept and update grounding status
4:   return
5: else
6:   inform the deliberative processing component
7:   update the grounding status of the corresponding object
8: end if

```

The generation of backchannel feedback is much more complicated than the interpretation of backchannel feedback and it is out of the scope of this thesis. However, to give a brief sketch of how the generation of backchannel feedback could be realized like in the framework

of FADE: Given an incremental speech processing chain, FADE would receive a continuous input of mentioned concepts. If the character represented by this instance of FADE has the role of an addressee and receives some discourse entity that is neither to be expected within nor related to the current topic of the conversation, FADE would need to prepare the generation of a negative backchannel feedback. However, what strength of backchannel feedback is presented depends on whether the speaker just requested such a feedback and this again is hard to recognize.

8.5.2 Identifying the Intended Addressee(s)

The task of identifying the intended addressee(s) is only relevant for dialogue systems with more than two participants and is not required for dyadic systems. As discussed in section 3.3.3, the intended addressee(s) are signaled in different ways by speakers. Some of the common addressing techniques are explicit while others are rather tacit and require some contextual reasoning in order to determine the intended addressee.

In our approach, the identification of the intended addressee is to some extent application specific and is thus realized by means of PATE rules. Nevertheless, in this section we will discuss the generic aspects of our algorithm for identifying the intended addressee (see section 9.2.5 for the actual implementation in VIRTUALHUMAN).

The general approach is based on a hierarchical set of restrictions:

Utterance contains vocative: If the utterance of the speaker contains a vocative, the participants that are denoted by the vocative are the addressees of that utterance. [Increases likelihood to 1]

Speaker mentions participant: If the speaker explicitly names a participant within the utterance (no vocative), this participant is most likely not the intended addressee. [Reduces likelihood to 0]

Direction of the speaker's gaze: The participant(s) the speaker is gazing at at the end of the utterance is likely to be the intended addressee (and next speaker). [Increases likelihood by 0.5]

Amount of speaker's gaze: The likelihood of being addressed increases based on the amount of gazing-time by the speaker during the turn. [Increase likelihood by (turn duration/individual time of gazing)]

Pointing gestures by the speaker: If the speaker points at other participants, the likelihood of these participants being the intended addressee(s) increases. [Increases likelihood by 0.3]

Speaker's body orientation: If the speaker changes his or her body orientation, the participants that are then faced by the speaker are more likely to be addressed. [Increase likelihood by 0.2]

Contextual factors: If none of the previous rules can be applied, the previous addressee is the current speaker and the dialogue act of the previous utterance is a subtype of Request, then take the previous speaker as addressee.

If no clear addressee could be identified, then consider all active, ratified participants as being addressed. This means that a subsequent component (i. e., the dialogue manager) has to determine whether it is appropriate to react or not.

8.5.3 Robust and Generic End-Of-Turn Detection

As discussed in section 4.3.4, a robust and fast end-of-turn detection is vital when aiming for a smooth and natural exchange of turns. In short, the problem is that since multimodal contributions are inherently composed of several monomodal events, the system can never be sure whether the last recognized event really marks the end of a turn. Thus, the system has to wait some time before it can stop the collection of input events and start with the interpretation process. Since recognizers (in particular speech recognizers) cause serious delays, this timeout has to be within the area of seconds.

In (Pfleger and Löckelt, 2005) we introduce an approach which is capable of reducing the gap between the end of a user turn and the subsequent system response. The key idea of this approach is to utilize the onset detection functionality of the modality specific recognizers. This means a recognizer sends a specific message to FADE every time it detects the beginning of an input event. Furthermore it will also send a message at the end of an input event in any case, even if the event appears to be a false alarm. This means FADE will receive two messages for any input event: (i) a message marking the onset of an input event and (ii) a message marking the end of that event. Since the first message will be sent with almost no delay, there is no need to wait for subsequent input events before a turn can be considered closed.

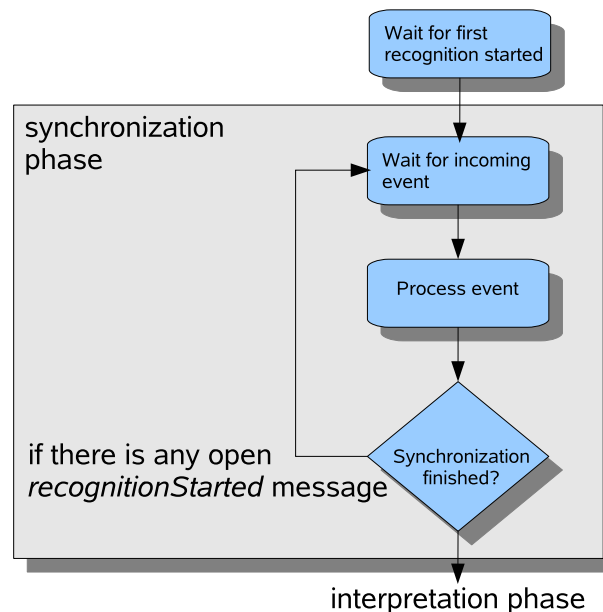


Figure 8.4: End-of-Turn detection and synchronization of incoming monomodal events.

Figure 8.4 depicts the basic processing steps for this approach to end-of-turn detection.

During the synchronization phase, all incoming events are pre-processed and analyzed as to whether they open a new expectation for a pending recognition result, or whether they close a previous *start-of-recognition* message. This processing strategy is realized by means of a set of PATE rules so that it is easy to adapt this mechanism to system specific formats of the start-of-recognition messages.

8.6 The Discourse Context

The purpose of the discourse context is to support and realize the processing of the context dependent phenomena that are discussed in Chapter 3. These phenomena are the following: (i) the resolution of elliptical and fragmental expressions, (ii) the resolution of referring expressions and (iii) the integration of information presented in different modalities into the discourse context. Since the overall design and organization of the discourse context of FADE is discussed in section 6.3, we focus here on the implemented features of the objects at the individual layers.

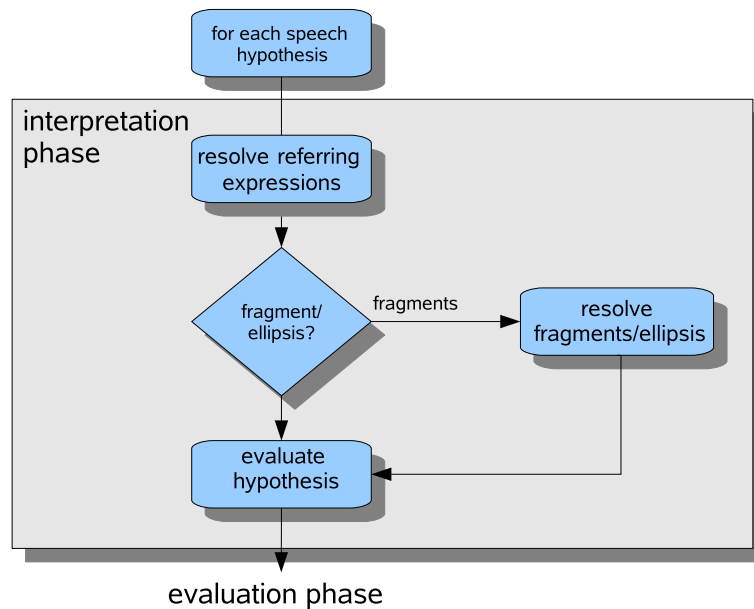


Figure 8.5: FADE's high-level processing strategy for the contextual interpretation of the propositional content of a user contribution.

The overall processing strategy for interpreting the propositional aspects of the user's contribution (called the *interpretation phase*) is organized as follows (see figure 8.5): For each hypothesis, first all embedded referring expressions are resolved (see below), then it is determined whether it is a fragment or ellipsis that needs to be resolved with the discourse context. Then, the resulting multimodal interpretation hypothesis is evaluated (see section 8.6.6). In a final step (called the *evaluation phase*) the individual hypotheses are ranked by using a discourse score that has been computed for each interpretation hypothesis.

First, we will briefly discuss the data model that is used to represent the objects at

the three layers of the discourse model. Then, we will discuss the processing strategies for reference resolution. The resolution of elliptical expressions is discussed in section 8.6.3 while the mutual disambiguation of monomodal interpretation hypotheses is discussed in section 8.6.6.

8.6.1 A Layered Multimodal Discourse Model

As discussed in section 6.3, the discourse model of FADE comprises three distinct layers inspired by the work of Luperfoy (1991, 1992). But as discussed in Chapter 6, our model shows some significant differences compared to Luperfoy's model. In the following sections we will outline our implementation of the discourse model comprising a *Modality Layer*, the *Referential Layer* and the *Discourse Layer* (see section 6.3).

The Modality Layer

The Modality Layer comprises three different types of objects: (i) Linguistic Actions (LAs), (ii) Nonverbal Actions (NAs) and (iii) Physical Actions (PAs). These objects represent the mentioning of objects at the Referential Layer (represented by the *is-anchored-by* relation in Luperfoy's model; see section 4.4.3) within the discourse and either introduce or reinforce an already introduced object at the Referential Layer. A linkage between an object at the Modality Layer and the Referential Layer also means that the activation of the object at the Modality Layer is transferred to the Referential Layer.

In the following, we will describe the individual features of the objects at the Modality Layer.

Linguistic Actions (LAs) Linguistic Actions reflect verbal references to objects at the Referential Layer. An LA models the specific characteristics of the actual surface realization. This information is used during the resolution process of referring expressions.

- *RealizationTime*—the time point at which the referring expression was uttered.
- *LexicalInformation*—the lemma used to reference a concept.
- *Number*—the number of the referring expression.
- *Gender*—the gender of the referring expression.
- *TypeOfReference*—defines the type of the referring expression: *definite*, *indefinite*, *deictic*.
- *IsAnchoredBy*—the link to the corresponding object at the Referential Layer.

Nonverbal Actions (NAs) Nonverbal Actions represent all kinds of nonverbal actions performed by the participants of a conversation that comprise propositional information.

- *RealizationTime*—the time point when the nonverbal action was performed.
- *Duration*—the length of time the nonverbal action was displayed.
- *Type*—type of the nonverbal action: *gaze*, *pointing gesture*, *iconic*, *emblematic*.
- *IsAnchoredBy*—link to the corresponding object at the Referential Layer.

Physical Actions (PAs) Physical Actions describe the appearance or disappearance of objects in the physical environment. They comprise information about the type of the event, when it happened and about the spatial properties of that object (including its relative position to other objects in the scene).

A Physical Action encompasses the following information:

- *RealizationTime*—the time point when the Physical Object first appeared.
- *Duration*—the duration the Physical Object was visible (if the object is still visible, the field remains empty).
- *IsAnchoredBy*—the link to the corresponding object at the Referential Layer.
- *Type*—describing the type of the Physical Action: *Appear* or *Disappear*.
- *AbsolutePosition*—a representation of the location of that object represented by an absolute position (see section 6.2.3).

The Referential Layer

The objects at the Referential Layer represent instances that have either been mentioned during the discourse, or that have been activated by some other means (i. e., through the physical context, or through the mentioning of associated instances).

Discourse Objects Discourse Objects are containers for instances that were directly mentioned during the preceding discourse. They comprise a unified representation of the semantic information gathered so far. In case a Discourse Object is completely resolved, its unified representation is replaced by a link to the corresponding instance of the LTM. Additionally, it contains a set of links to instances of the LTM (more than one in case of ambiguous/under-specified references) and links to objects at the Modality Layer; every time a Discourse Object is mentioned, a new link is added. Discourse Objects comprise the following information:

- *Activation*—representing the activation of the object.
- *ReferentialStatus*—the current referential status of the instance: *in focus*, *activated*, *familiar*, *uniquely identifiable*, *referential*, *type identifiable*.
- *Sponsor*—a link to the individual sponsor at the Modality Layer.
- *GroundingStatus*—representing the grounding status of this object: *introduced*, *grounded*.
- *UnifiedRepresentation*—holding the semantic representation in terms of ontological representation of the object.

Implicitly Activated Objects Implicitly Activated Objects are objects that are related to a Discourse Object. If a Discourse Object accesses an instance in the LTM, the activation of instances related to it is increased by a dynamic factor which depends on the activation of the superordinated instance and the *strength* of the relation between them. The spreading of activation is a recursive process (see section 6.4).

Implicitly Activated Objects may appear in the WM when their corresponding Discourse Object appears. This happens in case their activation exceeds the threshold. Also, the

activation of Implicitly Activated Objects decreases faster than that of Discourse Objects. Consequently, they are only accessible for a short time.

The represented features of an Implicitly Activated Object are:

- *Activation*—represents the activation of the object.
- *ReferentialStatus*—since the Implicitly Activated Objects have never been directly mentioned during the course of the interaction, they can only be: *uniquely identifiable*, *referential*, or *type identifiable*.
- *GroundingStatus*—represents the grounding status of this object: *associated*.
- *SemanticRepresentation*—the semantic representation in terms of an ontological representation of the object.

Physical Objects Physical Objects represent objects that can be perceived from the visual environment. If a Physical Object is explicitly activated through the mentioning of a Discourse Object, it can serve as a referent for a referring expression. Physical Objects are not only part of the Referential Layer, as they are also part of a superordinate structure representing the complete physical surroundings by modeling the relations between the physical objects located in a scene (e. g., the grey building is to the left of the blue building).

The represented features of a Physical Object are:

- *Activation*—representing the activation of the object.
- *ReferentialStatus*—the current referential status of the instance: *uniquely identifiable*, *referential*, or *type identifiable*
- *GroundingStatus*—representing the grounding status of this object: *visible*, *nonvisible*
- *SemanticRepresentation*—the semantic representation.

The Discourse Layer

The primary purpose of the discourse layer is to model the sequence of utterances of the individual participants (like the sequence memory of the Verbmobil discourse model; see Alexandersson (2003)). To this end, the discourse layer encompasses a turn-based representation of the evolving discourse. For each turn, it stores information about the speaker, the semantic representation of the content of that utterance including the dialogue act as well as temporal information. The discourse layer is updated every time a participant finishes a contribution.

Consider, for example, the following excerpt from a sample interaction with the VIRTUALHUMAN system:

- (86) **Moderator:** [gazes at Ms. Herzog] “*Ms. Herzog, what do you think?*”
- (87) **Herzog:** [gazes at the moderator] “*Well, I think the goalkeeper saves the ball.*”
- (88) **Moderator:** [gazes at User 1] “*Ok, User 1, what do you think?*”
- (89) **User 1:** “*Hhm, I think Ms. Herzog is right.*”

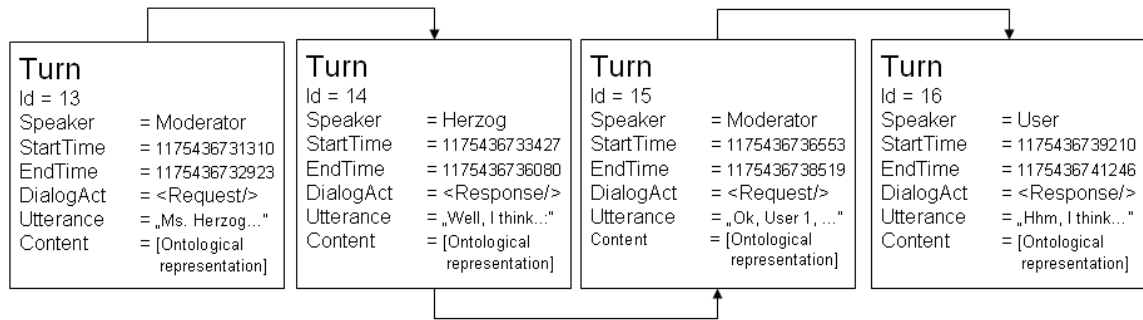


Figure 8.6: Example configuration of a turn-sequence of four turns involving three speakers.

This interaction would result in a configuration of the Discourse Layer as depicted in figure 8.6. Using this sequential representation it is, for example, possible to resolve discourse references that refer to a contribution of an arbitrary previous speaker. Besides the sequential representation of the previous discourse, FADE also supports the representation of adjacency pairs and the thematic organization of the previous discourse based on so-called focus spaces (see Pflieger (2002); Pflieger et al. (2003a,b)).

8.6.2 Resolution of Referring Expressions

The resolution of referring expressions is realized in FADE as a sequential process that operates on the Referential Layer of the discourse model representing the current discourse context.

As discussed in section 3.2.3, there are referring expressions that can be classified context-free, i. e., without considering the previous discourse, and there are also others that cannot be classified context-free. If the actual type of the referring expression cannot be determined unambiguously, the reference resolution algorithm tests several hypotheses during each run. Table 8.1 shows the resolution strategies that are supported by FADE.

Viewed on an abstract level, the algorithm operates as follows: Given an unambiguous referring expression, the algorithm traverses the objects at the Referential Layer until a match is found. In case of an ambiguous referring expression, the algorithm traverses the objects at the Referential Layer several times. In each run it adopts a different hypothesis about the type of referring expression and tries to resolve it. Only if there was no matching object, is a second or possibly third run through the list started with a different resolution strategy. This search continues until the algorithm encounters an object that satisfies the resolution criteria of the current hypothesis. Finally, if no matching object has been found, there is an additional run assuming a discourse reference which is resolved by traversing the sequence of previous utterances by the individual participants.

FADE's API for resolving referring expressions provides a single method called *resolveReferringExpressions* that encapsulates all specialized resolution methods. When called, this method first analyzes the referring expression and tries to determine its type. Then, it applies the specific resolution strategy for that type. The actual resolution process is realized through a number of specialized resolution methods which will be discussed later in this section:

Type of reference	Examples	Resolution strategy
Noun-phrases		
Proper nouns (Homophorics)	<i>John, New York, Germany, the Queen, ...</i>	1) resolve anaphoric reference 2) resolve exophoric reference 3) resolve via LTM
Common nouns (+ definite article)	<i>the car, the song, the movie, etc.</i>	1) resolve total anaphoric reference 2) resolve discourse anaphoric reference 3) resolve indirect reference
Common nouns (+ demonstrative)	<i>this car, this song, this movie, etc.</i>	1) resolve place deictic reference 2) resolve exophoric reference 3) resolve anaphoric reference
Common nouns (+ ordinal)	<i>the first song, the last movie, etc.</i>	1) resolve reference to collection
Common nouns (+ adjective)	<i>the red cable, the old car, etc.</i>	1) resolve exophoric reference 2) resolve anaphoric reference
Pro-Forms		
Pronouns	<i>it, he, she, etc.</i>	1) resolve anaphoric reference
Demonstratives	<i>this, that, these, etc.</i>	1) resolve place deictic reference 2) resolve anaphoric reference
Temporal references	<i>tomorrow, tonight, now, etc.</i>	1) resolve temporal reference
Spatial references	<i>here, there, etc.</i>	1) resolve spatial reference

Table 8.1: Overview of the different strategies for the resolution of different types of referring expressions.

`resolveDeicticExpression`—resolves a place deictic reference and integrates it with accompanying gestures.

`resolveAnaphora`—resolves anaphoric references.

`resolveDiscourseAnaphora`—resolves discourse anaphoric references.

`resolveTemporalDeixis`—resolves temporal references.

`resolveReferenceToCollection`—resolves a referring expression that refers to a specific entry in a collection (e. g., lists displayed on the screen or spoken enumerations).

`resolveExophoricReference`—resolves exophoric (cross-modal) references.

`resolveSpatialReference`—resolves spatial references.

`resolveIndirectReference`—resolves indirect references.

`resolveDiscourseDeicticExpression`—resolves discourse deictic references.

Before the individual resolution methods will be discussed in detail, we will first introduce the representation format for referring expressions.

Representation of Referring Expressions

FADE requires the speech analysis component to extract as much information as possible from the intra-sentential context to support the process of reference resolution. First of all, it is very important to use the semantic restrictions posed by the verb to determine the basic meaning of the referring expression. Consider for example the following utterance:

(90) “*Record it.*”

Here the referring expression denotes some entity that can be recorded which drastically reduces the set of potential referents. At this point, it is possible to utilize the type-hierarchy of the ontology which might comprise higher level concepts that subsume the types of the potential referents. For our example we could use a class called *AVMedium* subsuming all concepts that can be recorded (e.g., broadcasts, movies, songs). The more specifically the semantic class of the referring expression can be inferred, the smaller the set of potential referents gets.

Secondly, FADE needs information about the type of referring expression. We distinguish four types of referring expressions that can be assigned without access to the context:

indef—an indefinite reference to a concept, e.g., *a car*.

def—a definite reference to a particular entity, e.g., *the car*.

deictic—a deictic expression, e.g., *this car*

trueDeictic—a deictic expression that requires an accompanying pointing gesture, e.g., *this car here*.

Thus, a completely analyzed referring expression *this car* looks like this:

```
<Car>
  <has_lingInfo>
    <LingInfo>
      <has_refProp>
        <RefProp>
          <has_type> deictic </has_type>
        </RefProp>
      </has-refProp>
    </LingInfo>
  </has_lingInfo>
</Car>
```

A more complex example is the referring expression *the third car*:

```
<Car>
  <has_lingInfo>
    <LingInfo>
      <has_refProp>
        <RefProp>
          <has_type> deictic </has_type>
          <has_differentiationCriterion>
            <DifferentiationCriterion>

```

```

      <has_DCValue> third </has_DCValue>
    </DifferentiationCriterion>
  </has_differentiationCriterion>
</RefProp>
</has-refProp>
</LinfInfo>
</has_lingInfo>
</Car>

```

Finally, as has been discussed in section 3.2.3, syntactic constraints like number, person and gender usually help to identify the referents. Thus, FADE also needs syntactic information for the referring expressions and for potential referents. To this end, the speech analysis module is required to add syntactic information to the analysis result if available⁸. This syntactic information should be embedded as in the following example:

```

<Car>
  <has_lingInfo>
    <LingInfo>
      <has_refProp>
        <RefProp>
          <morphoSyntacticDecomposition>
            <WordForm>
              <number> singular </number>
              <gender> male </gender>
            </WordForm>
          </morphoSyntacticDecomposition>
          <has_type> def </has_type>
        </RefProp>
      </has-refProp>
    </LinfInfo>
  </has_lingInfo>
</Car>

```

Resolving Place Deixis/Multimodal Fusion

The resolution of place deixis is the point where the actual fusion of two modalities takes place. As we have discussed in section 4.3, this is typically handled by a separate component. In our approach, however, the integration of two modalities is done during the reference resolution.

Generally, we distinguish two types of place deixis: (i) place deixis without accompanying pointing gestures and (ii) place deixis with accompanying gesture. A place deictic referring expression without accompanying pointing gesture (e. g., *this* or *that*) is typically ambiguous since it can be interpreted either as a place deixis to a visually distinct entity, as an anaphoric reference to a linguistically distinct entity, or as a reference to the discourse.

Examples of the second category, i. e., a place deictic reference that is accompanied by pointing gestures, looks like this:

this/that + pointing gesture (hand/pen/gaze)

here/there + pointing gesture (hand/pen/gaze)

these/those + encircling gesture (hand/pen/gaze)

⁸In case the speech analysis module does not provide syntactic information, the corresponding consistency check (see the next section) cannot be employed. This means that the quality of the resolution results will be comparatively lower.

these/those + a series of pointing gestures (hand/pen/gaze)

As discussed in section 2.3.2, there are also other types of combined multimodal input as illustrated by the following examples:

Iconics: “*It’s no longer than this*” while displaying an iconic gesture.

Drawings with the pen: “*Its length is 2m*” while drawing a window into a blueprint.

Iconics and drawings are processed just like pointing gestures as their semantic representation and temporal characteristics need to be compatible with the verbal reference.

The actual processing of multimodal references is organized as follows: First, the list of accessible objects at the Referential Layer is traversed looking for Discourse Objects that have been referenced in the same turn by some kind of a pointing gesture or iconic gesture. For each of the identified Discourse Objects, the algorithm determines whether the referring expression and the pointing gesture can be integrated by applying the following three constraints:

Temporal consistency check: The surface realization time of the gesture must be compatible (i. e., similar) to the realization time of the referring expression. Note that it is not tested for an exact match as people tend to show sequential integration patterns (see Oviatt et al. (2003)).

Syntactical consistency check: The cardinality restrictions of the referring expression and the pointing gesture must be compatible. If the referring expressions requires for multiple objects (as for instance in “*show me these cars*”), this consistency check would only accept an encircling gesture that contains more than one object.

Semantic consistency check: The semantic content of the object referred to by the gesture and the linguistic expression must be compatible (i. e., unifiable).

In case of a plural referring expressions (i. e., “*these*”), that could not be resolved with an encircling gesture, the algorithm runs a fallback strategy that assumes multiple pointing gestures selecting a set of objects subsequently.

Resolving Anaphoric References

Determining whether a referring expression is anaphoric and to resolve it then is comparatively straightforward. As we have seen in section 3.2.3, an anaphora must be linked to the element of the prior discourse to which it is “pointing”. Both the anaphora and the antecedent are referring expressions and in the case of total anaphora both refer to the same *referent* in the *real world*.

The identification of the intended referent for an anaphoric reference takes place by applying the following constraints to the individual Discourse Objects at the Referential Layer:

Syntactical consistency check: The referring expression must have the same number and gender information as the Linguistic Action when the Discourse Object was last mentioned.

Semantic consistency check: The semantic content of the Discourse Object and the referring expression must be compatible (i. e., unifiable).

As the Discourse Objects at the Referential Layer are always ordered with respect to their activation, the first match is always the best. Thus, the algorithm stops when it encounters a compatible Discourse Object.

Resolving Discourse Anaphoric References

The resolution of discourse anaphoric references basically resembles the resolution of total anaphoric references, with the difference, however, that in this case only the semantic consistency check is applied.

Resolving Temporal Deixis

The resolution of temporal references depends, like the resolution of place deixis and spatial references, on a currently active frame of reference. This frame of reference is called the *temporal* frame of reference and represents a definite date and time that is either the current date and time or has been previously mentioned in the discourse. The latter case is often linguistically marked as in “*Let’s not meet on Monday but three days later.*”

Temporal reference resolution determines what time and date is being specified by temporal referring expressions (e.g., “*tomorrow*” which has to be resolved to a concrete date, e.g., *Saturday, the 5th of August*). However, the interpretation of tense and aspect would involve the determination of implicit information about the states and events specified by verb phrases (e.g., that the event specified in “*He had crashed the car*” happened before the time stated by the current frame of reference) and this is outside the scope of this thesis.

We distinguish two types of temporal referring expressions: (i) *relative* temporal references, i.e., those that depend on the current temporal frame of reference and (ii) *absolute* temporal reference, i.e., those that do not need to be interpreted but change the current temporal frame of reference.

Resolving Temporal Expressions Temporal expressions have to be encoded according to the meta-ontology of FADE. They can be embedded within a structure modeling a single time point or a time interval. Time intervals (e.g., “*next week*”) will be partitioned into two time points (e.g., “*next week*” would result to a temporal interval consisting of the time points: *Monday* and *Sunday*) which will be processed one after the other.

The algorithm for resolving relative temporal references has two arguments: (i) the current active temporal frame of reference and (ii) the temporal reference. The actual resolution is rather simple: First, the referring expression is decomposed into its components concerning the date and the time (e.g., *tomorrow night*). Note that some temporal references contain only one of these components (e.g., *tomorrow, evening*). Then, it is determined whether the temporal expression refers to the current time frame of the current temporal frame of reference.

The processing of the components that concern the date is organized as follows: First it is determined whether the referring expression expresses a deictic reference (e.g., *tomorrow*) or a calendric day (e.g., *Saturday*). For deictic reference, the algorithm first looks up the offset in days in a mapping table. For instance, the deictic reference *in three days* would be mapped to an offset of 3 (table 8.2 gives some examples of referring expressions and their offset).

As discussed in section 3.2.3, time deictic references like “*tonight*” or “*afternoon*” can sometimes be vague and require domain specific knowledge. However, since FADE is designed

Absolute Temporal References	Offset from <i>now</i>
<i>yesterday</i>	- 1 day
<i>tomorrow</i>	+ 1 day
<i>the day after tomorrow</i>	+ 2 day
<i>today</i>	0 days
<i>in three days</i>	+ 3 days
...	
Relative Temporal References	Offset from active frame-of-reference
<i>that day</i>	+ 0 days
<i>an hour later</i>	+ 1 hour
<i>the day thereafter</i>	+ 1 day
...	

Table 8.2: Examples of temporal deictic expressions and the corresponding offset in days or hours.

as domain and application independent as possible, we decided to use fixed intervals for these vague expressions (e. g., *tonight* or *evening* will be mapped to 6 pm - 11:59 pm). All other time deictic references will be processed just like the date deictic references. A mapping table (see table 8.3 for a short excerpt) provides offset information for the common absolute and relative temporal references which can be resolved by FADE.

Referring expression	offset in hours:minutes
two hours later	2:00
two hours earlier	-2:00
half an hour later	0:30
...	

Table 8.3: Examples of time deictic expressions and the corresponding offset in hours and minutes.

Resolving Holidays Special days or Holidays are subject to legal regulation of the individual countries so that the number, names and actual dates of holidays can vary. Additionally, many holidays are *dynamic*, i. e., they are not bound to a fixed date like New Year's Eve but are defined relative to the Easter date of the respective year. To this end, FADE comprises a subcomponent that is able to compute the date of all relevant holidays of a specified country.

This subcomponent computes the date of a holiday or special day in two steps: (i) first it looks up the date in an XML-database for holidays, and (ii) if the holiday is a dynamic date, it computes the actual date. This XML-database comprises information for each common holiday or special day, for example, its names (many holidays have more than one name), its type (*fixed* or *dynamic*) and depending on this type, the actual date (in case of a fixed day) or its distance relative to Easter (in case of a dynamic date). Figure 8.4 shows a simplified excerpt of this database.

The computation of the actual dates of dynamic holidays is based on the *Easter* algorithm (see Tondering (2005); last visited August 2006). This algorithm computes the exact day of

Name	type	date	distance to Easter
Neujahr	fix	01/01	
Mariä Himmelfahrt	fix	08/15	
Allerheiligen	fix	11/01	
Tag der Deutschen Einheit	fix	10/03	
Ostersonntag	dynamic		0
Ostermontag	dynamic		1
Palmsonntag	dynamic		-7
Karfreitag	dynamic		-2
Pfingstsonntag	dynamic		49
...			

Table 8.4: Some dates for dynamic and fix holidays in Germany.

Easter based on the Gregorian calendar. Given the Easter date and the distance of a day to Easter, it is easy to compute the actual date of a dynamic holiday.

Resolving References to Collections

References to collections like, for example, “*Play the first song.*” or “*Activate the last one.*” can refer either to collections presented on the display (e. g., lists or tables) or to collections introduced by speech (e. g., “*The first four songs on the album “One” of the Beatles are: ‘Love me do’, ‘From me to you’, ‘She loves you’ and ‘I want to hold your hand’.*”). In both cases, the user can use a referring expression like “*Play the second song.*” in order to refer to the second song of the collections.

For the resolution of referring expressions that refer to a specific entry of a collection, FADE needs first to determine the collection with the highest activation as it is the most likely sponsor. Then the list of embedded objects is traversed and the semantic consistency check is applied to each object. The first match is returned as the result of the reference resolution process. If no matching referent has been found, the algorithm continues with the next collection. However, this is done only for collections that have been recently mentioned or that are still displayed on the screen.

Resolving Exophoric (Cross-Modal) References

As discussed in section 3.2.3, exophoric or cross-modal references are references to the visual or situational context of the discourse. Like other types of referring expressions, this type of reference cannot always be classified unambiguously as an exophoric reference. However, because of the ordered sequential processing strategy for referring expressions, we already eliminated a number of hypotheses regarding the type of the referring expression when the method `resolveExophoricReference` is called. This means that the actual process of identifying the indented referent can be realized by means of a rather simple search through the objects present in the physical environment (i. e., the Physical Objects at the Referential Layer) . If this search reveals a semantically compatible object, it is clear that the referring expression must be an exophoric one. Otherwise, we need to proceed with the next step in the resolution hierarchy.

Again, semantic compatibility is tested by iteratively unifying the referring expression with the objects present in the physical environment, starting with the object with the highest activation.

Resolving Spatial References

Resolving spatial references depends on the point of view the speaker takes to encode the referring expression which is called the *frame of reference* (Levinson, 2003; see section 3.2.3). The frame of reference a speaker takes, directly influences the selection referring expression, e. g., everything that is on my left is on the right of someone else facing me. The structure and distinctions between frames of reference has been a research topic in various disciplines such as philosophy, cognitive science, psychology and linguistics. Of course this led to various conceptions and terms describing different types of frames of reference. Based on a detailed review of these different notions of frames of reference, Levinson distinguishes three main frames of reference: *intrinsic*, *relative* and *absolute* (Levinson, 2003, p.35). When using an intrinsic frame of reference, the speaker takes the point of view of the relatum (i. e., the object that is used to locate the target object). In a relative frame of reference, the speaker takes an outside perspective (e. g., his own point of view, or that of someone else). Within an absolute frame of reference, everything is located with respect to the geographic north. While the latter frame of reference is always unambiguous, the former two might introduce some ambiguities that need to be resolved.

The resolution of spatial reference involves the following sources of information: (i) an up-to-date representation of the physical environment, (ii) knowledge of the currently active type of frame of reference and (iii) a mapping function that converts spatial references to locations or objects in the scene.

The representation of the physical environment is organized as follows (see also section 6.2.3): Each object located in the scene is represented by means of an *AbsolutePosition*. Moreover, a physical environment can again contain other physical environments as, for example, in the VIRTUALHUMAN scenario. There, the physical environment that describes the studio contains a closed physical environment which describes in turn the virtual football field where the line-up is displayed (another example would be a physical environment that describes a house that consists of several rooms which again are closed physical environments). The most important features of this structure are:

- Feature *ontologicalInstance*—this feature contains the ontological instance representing the object that is described by an *AbsolutePosition*.
- Feature *coordinates*—this is an optional feature; if the perception components are able to provide two-dimensional or three-dimensional coordinates describing the location of the object with respect to a fix coordinate system, this slot will contain that information.
- Feature *orientation*—this feature describes the current orientation of the object; valid values are: *north*, *east*, *south*, *west*.
- Feature *northOf*—this feature contains a link to the *AbsolutePosition* of its northern neighbor (can be empty if there is no northern neighbor)
- ...

This means that each `AbsolutePosition` that represents an object in the scene also represents the spatial relations the object currently fulfills with respect to the organization of the scene from an absolute point of view (using viewpoint neutral descriptions; see figure 8.7 for an example configuration of the physical environment).

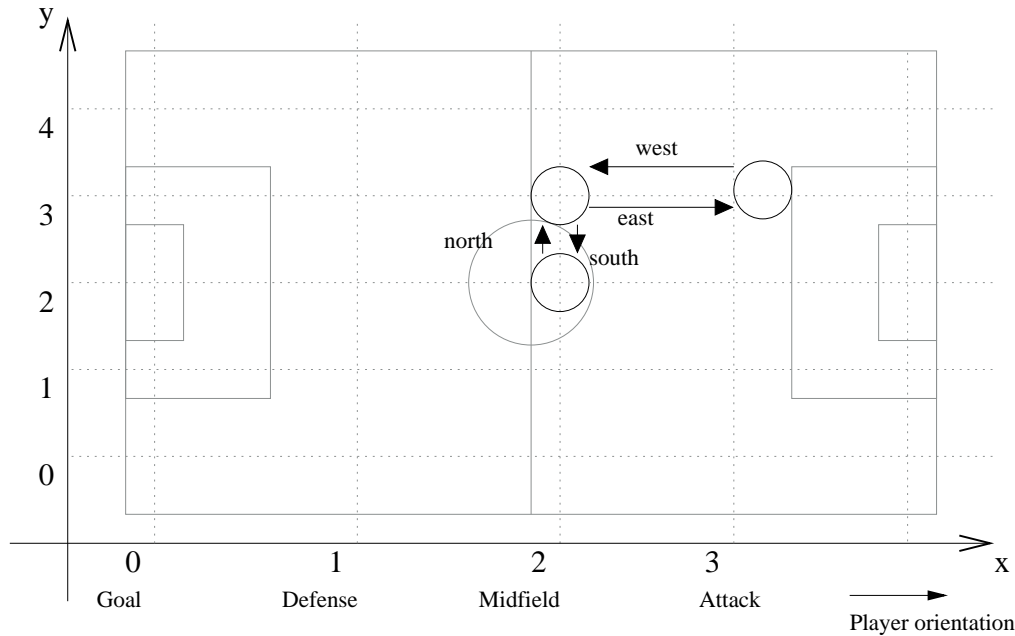


Figure 8.7: Example configuration of the physical environment describing the football field with three players placed on it.

In order to resolve spatial references, FADE first determines the currently activated physical environment and its corresponding active frame of reference and then maps the referring expression to an absolute location. If, for example, the user commands the system to “Place Ballack to the left of Klose”, the system first searches for the current position of the player *Klose* (which means it resolves the exophoric reference to the object representing the player *Klose*) in the physical environment. Then it retrieves the current orientation of that player and maps the referring expression to one of the absolute identifiers.

At this point we assume a currently active frame of reference of type *intrinsic*, otherwise the system would need to determine the orientation of the speaker and then compute the mapping. In any case, the mapping function takes the referring expression (*left-of*) and the orientation of the relatum (*eastern*) which would result to an offset of 1. This means we need to go *one* neighbor feature further on to get the correct neighbor given the orientation. Normally (i. e. if the player is oriented to the north), left-of would be mapped to the western neighbor. In our case, however, we need to go one neighbor further, which turns out to be the northern neighbor. If the player faces westwards, the mapping function would return an offset of 3 which means left-of now becomes the southern neighbor.

Resolving Indirect References

Indirect references are resolved by means of the Implicit Objects that are currently accessible in the Referential Layer⁹. Implicit Objects, as discussed in the previous section, represent instances from the LTM that received an activation through the spreading activation mechanism (see section 7.2.3). In order to resolve this type of reference, the algorithm loops over the Implicit Objects that are currently present at the Referential Layer starting with the object with the highest activation. The first Implicit Object satisfying the semantic consistency check will be returned. Consider, for example, the following the sample interaction with the SMARTWEB system:

- (91) User “*How did the game Germany against USA during the World Cup 2002 end?*”
 (92) System 1:0.
 (93) User “*How often did Ziege score?*”

In this example the user’s first question sets the context for the interpretation of the second one. A possible context-free interpretation of the second contribution is “*How many goals did Ziege score in his entire career?*”. However, in the given context, the intended interpretation is rather “*How many goals did Ziege score in this particular game?*”.

The intended interpretation is settled by the fact that the football player “Ziege” was a member of the German team and participated in that particular game. Thus, what we need is to establish a link between the instance representing the FOOTBALLPLAYER Ziege, which is embedded in an under-specified FIELDMATCHFOOTBALLPLAYER and the already activated instance of the class FIELDMATCHFOOTBALLPLAYER of the game mentioned in the previous user turn. This link can only be established if not only the game itself is activated but also related instances like all the players that participated in that game and the location where the game took place are activated as well.

Resolving Discourse Deixis

As discussed in section 3.2.3, discourse deixis or discourse references refer to an entire utterance in the previous discourse. Discourse references can be realized in various ways, as for example in the following way:

- (94) “*I like **that**.*”
 (95) “*Let’s do **that**.*”
 (96) “*Mr. Kaiser is right.*”

While the first two examples ((94) and (95)) contain an actual referring expression, the discourse reference in example (96) is realized without a referring expression. The resolution of discourse references is rather straightforward, however, it partially depends on the actual representation of dialogue acts. To this end, the resolution of discourse reference consists of two steps: (i) the retrieval of the referenced utterance (this is the generic part) and (ii) the integration of the semantic interpretation of the utterance into the discourse reference (the non-generic part).

⁹See also Pflieger and Alexandersson (2006) for a more detailed discussion of the resolution of indirect reference.

On the implementation side, the discourse context of FADE provides a method called `retrieveUtterance(speaker)` for retrieving the previous utterance of a particular speaker. The integration is realized by means of PATE rules.

8.6.3 Resolving Elliptical Expressions and Fragments

As discussed in section 3.2.4, the resolution of elliptical expressions in dialogues requires an integration of the fragment into the frame provided by either the utterance of the previous speaker or the previous utterance of the current speaker.

FADE supports two different approaches for the resolution of elliptical expressions: (i) the resolution solely based on the discourse context and (ii) the resolution solely based on expectations generated by the dialogue manager. In the following subsections we will describe the two approaches in more detail.

Resolving Ellipsis Based on the Discourse Context

As discussed in section 3.2.4, discourse ellipses and elliptical answers to questions are among the most frequent fragmental utterances (as analyzed by Schlangen (2003); Schlangen and Lascarides (2003)). These are the two primary elliptical constructions FADE is able to deal with.

An important aspect for the determination of the correct sponsor utterance is the interaction pattern as exemplified by the following interaction patterns:

- (97) **A:** Question -> **B:** elliptical Answer
- (98) **A:** Question -> **B:** (elliptical) Answer -> **A:** elliptical Question
- (99) **A:** Statement -> **A:** Statement

FADE's API provides a single method for resolving elliptical constructions that takes a fragment as argument and returns, if possible, a complete dialogue act representing the resolved fragment. Before the actual resolution takes place, FADE first needs to determine the interaction pattern. If the previous turn was a question of a different speaker, the algorithm assumes an elliptical answer to a question and processes the fragment accordingly (see below). Otherwise it assumes a discourse ellipsis and processes the fragment in the following way.

Resolving Discourse Ellipsis Discourse ellipsis usually occurs in follow-up questions where speakers refine or correct their previous question, as in the following discourse fragments:

Replacing a previously set slot:

- (100) **User:** *"What's the weather going to be like today?"*
System: *"..."*
User: *"And tomorrow?"*
- (101) **User:** *"What's happening in Saarbrücken today?"*
System: *"..."*
User: *"And in Karlsruhe?"*

What can be observed is that this types of elliptical expressions is characterized by a type identity between the elliptical expression and the replaced part of the sponsor sentence. In example (100), the replaced part as well as the fragment are temporal expressions (i. e., instances of the type `TimePoint`). The same holds for example (101). Here, fragment and the replaced part of the sponsor sentence are both towns.

In order to be able to resolve such elliptical constructions, FADE searches in semantic representation of the possible sponsor utterances for an instances of the same concept as the elliptical expression. If a matching instance is found, FADE overlays the fragment and its counterpart. The resulting structure is then incorporated into the semantic representation of the previous utterance. Note that this approach also supports the resolution of expansion ellipsis (see section 3.2.4) if the underlying ontological model supports an superclass–subclass relation between the instance mentioned in the elliptical expression and the replaced constituent of the previous utterance.

Resolving Elliptical Answers To Questions For the resolution of elliptical answers to questions, FADE needs additional information about the frame within which the fragment needs to be interpreted. Consider as an example the following question:

- (102) **User:** “*Show me the way from Stuttgart to...*” [the system fails to recognize the last word of the utterance]
System: “*Where do you want to go?*”
User: “*To Munich.*”

Since a typical route-planning concept in an ontology comprises a source and destination slot, it is important for FADE to know which of these slots is called for. To this end, FADE needs a pointer into the semantic representation of the questioned concept in order to be able to integrate the fragment. The actual integration then only consists of a consistency check that determines whether the fragment is semantically compatible (i. e., unifiable) with the target spot and then the system replaces the target spot with the semantic representation of the elliptical expression.

Expectation-based Fragment Resolution

Some dialogue managers are able to generate expectations about likely user input after each system turn. This is an approach that we originally developed for the SMARTKOM system (see Löckelt et al. (2002)). The overall idea is that the action manager generates the following set of expectations:¹⁰

- *expected*—describing the semantic representation of the most likely user input (there must be no more than one expected input).
- *possible*—describing the semantic representation of likely objects (there may be any number of possible objects).
- *filled*—containing the semantic representation of the objects that have already been provided by the user.

¹⁰Note that this approach only works for task-oriented dialogues where the user has to provide specific information in order to be able to continue the dialogue.

Consider, as an example, the following expectation (taken from the second version of the OMDIP demonstrator) which is triggered when the system asks for a telephone number and it is also possible to enter a time-point:

```
<Expectation>
  <is_possible>
    <Inform>
      <has_content>
        <DeliveryDate>
          <has_timePoint>
            <smartdolce:time-point/>
          </has_timePoint>
        </DeliveryDate>
      </has_content>
    </Inform>
  </is_possible>
  <is_expected>
    <Response>
      <has_content>
        <MusicGreetingMessage>
          <has_recipient>
            <PhoneNumber>
              <has_value>
                <Text/>
              </has_value>
            </PhoneNumber>
          </has_recipient>
        </MusicGreetingMessage>
      </has_content>
    </Response>
  </is_expected>
</Expectation>
```

Here, the user is expected to provide a *phone number*, however, he or she also has the possibility to specify a time point when the *music greeting* should be sent out.

The actual resolution of fragmental input is straightforward: First, FADE tries to integrate the fragment into the expected object. If this is not possible, the next integration step tests the individual possible slots. Eventually, if still no match can be found, FADE tries to integrate the fragmental expression into one of the filled objects. Otherwise, it is not possible to resolve the elliptical expression and it will be passed on to the dialogue manager.

8.6.4 Updating the Discourse Context

The discourse context of FADE needs to be updated every time a participant of the conversation finishes an utterance. During this update all mentioned Discourse Objects need to be integrated into the discourse context model.

How a Discourse Object is integrated into the discourse context depends on whether it has already been mentioned in the previous discourse. In case the Discourse Object does not exist, it is simply added to the discourse context (with an activation of 1.0). Otherwise, the existing Discourse Object is updated and its activation is also set to 1.0. The identity between two Discourse Objects is determined by unifying their semantic content that is embedded in the field *unified representation*.

In the final post-processing step, the activation of all Discourse Objects of the discourse context that were not mentioned during this turn is reduced. Then, the entire discourse context is sorted and all Discourse Objects are removed whose activation is below the threshold

that distinguishes the working memory from the long-term memory. All other Discourse Objects simply disappear.

8.6.5 Retrieving the Referential and Grounding Status of Referential Objects

As discussed in section 3.2.5, common ground plays a crucial role for a successful communication. To this end, FADE provides an interface for external components (i. e., the multimodal generator of VIRTUALHUMAN) that gives access to the referential and grounding status of objects at the Referential Layer. FADE computes the grounding status of these objects based on the referential status of a Discourse Object and the total number of mentionings during the discourse. The following referential statuses are distinguished:

- **New**—A Referential Object that has not been mentioned during the discourse so far.
- **Introduced**—A Referential Object that was mentioned during the previous discourse but that is not currently focused.
- **Focused**—A Referential Object that is currently in focus.

With respect to the grounding status of an object, FADE distinguishes three different states:

- **Ungrounded**—There is no grounding information available
- **Implicitly Grounded**—The Discourse Object was mentioned and no negative feedback has been received so far (this state is assigned if no negative feedback is received in the subsequent turn after the introduction of a Discourse Object).
- **Explicitly Grounded**—The Discourse Object was grounded by an explicit grounding act. The information of whether a Discourse Object is explicitly grounded needs to be provided by an external component (usually the dialogue manager).

8.6.6 Disambiguation of Multiple Interpretation Hypotheses

Each recognizer of a multimodal dialogue system possibly produces a set of interpretation hypotheses where each hypothesis stands for an independent and diverse interpretation of the input signal. A speech recognizer, for example, encodes this ambiguity typically by means of word lattices where several paths through the word lattice reflect the individual interpretations or n-best lists where each entry reflects a complete interpretation. But also a pen-input recognizer can produce a set of alternative recognition hypotheses, for example, if the user points in-between two objects (see Wahlster (1991); Wasinger (2006)). Typically, the recognizers also produce confidence scores that rank the individual hypotheses.

Disambiguation of multiple interpretation hypotheses in FADE takes place at two levels: (i) on the semantic level where the cross-product of the received interpretation hypotheses might result in a reduced number of semantically compatible hypotheses and (ii) by means of a scoring mechanism that reflects the ratio of unresolved references of the individual hypotheses.

The actual processing strategy of the individual hypotheses does not differ significantly from the processing strategy for unambiguous hypotheses. The most important part is the

collection of parameters that are used for the evaluation of the hypotheses at the end of the processing. Each interpretation hypothesis is processed in the standard way, i. e., the hypothesis is routed through the individual resolution stages. Then, a so-called *hypothesis evaluation* score is computed that is based on four parameters:

(SPEECH) is the score of the speech recognition and/or speech analysis component for that particular hypothesis (this score has to be in the codomain $[0, 1]$).

(NDR) is the ratio of unresolved non-deictic references, i. e., the number of resolved non-deictic referring expressions divided by the total number of non-deictic referring expressions (NDR is set to 1.0 if there was no non-deictic referring expression)

(DPG) is the ratio of integrated pointing gestures: The number of integrated deictic gestures divided by the total number of recognized deictic gestures (UPG is set to 1.0 if there was no non-deictic referring expression)

(UDR) is the ratio of resolved spatial-deixis: The number of resolved deictic references divided by the total number of spatial deictic references (UDR is set to 1.0 if there was no deictic referring expression)

Definition 13 (Fusion Score)

$$Score = \frac{SPEECH + NDR + DPG + UDR}{4}$$

This means that any unresolved references or any pointing gestures that could not be integrated will cause a reduction of the overall fusion score. But if there is no reference at all or all references are resolved, the score will remain the same as the score computed by the speech recognition or speech analysis component. Eventually, the algorithm selects the best-scored hypothesis which will then be passed on to the dialogue manager.

Consider, for example, the following simplified situation (taken from the OMDIP scenario): The user has uttered “Activate this Soundlogo” [plus an accompanying pointing gesture]. The speech recognizer, however, generates two hypotheses: (H1) “Activate the first Soundlogo.” with a confidence of 0.90 and (H2) “Activate this Soundlogo.” with a confidence of 0.85. The fusion score for these two hypotheses look like the following¹¹:

H1

$$score(H1) = \frac{0.90 + 1.0 + 0 + 1.0}{4} = 0.725$$

H2

$$score(H2) = \frac{0.85 + 1.0 + 1.0 + 1.0}{4} = 0.975$$

This means FADE would rearrange the hypotheses so that hypothesis H2 would be the preferred hypothesis.

¹¹Note that we assume that the spatial reference “*the first*” of H1 can be resolved via the physical context, otherwise the score for H1 would even be worse.

8.6.7 Contextual Enrichment and Validation

Besides the resolution of elliptical and referring expressions, FADE's API provides additional means for enriching a perceived utterance with contextual information. The basic idea of this contextual enrichment is to overlay (see section 5.2.4) the analysis result of a current utterance with the analysis result of a previous utterance of the same participant. The resulting interpretation for the current utterance then encompasses all compatible information of the previous one. This approach was developed for the discourse modeler (see Pflieger (2002); Pflieger et al. (2003a); Alexandersson (2003)) of the SMARTKOM system. Besides the contextual enrichment of interpretation hypotheses, this approach also provides a score reflecting how well an interpretation hypothesis fits its context. This score is called *validation* and it is computed during the application of overlay as discussed in section 5.2.4.

FADE's API provides a method called `contextualEnrichment` that takes an interpretation hypothesis and enriches it with compatible contextual information. Additionally, this method also updates the hypothesis evaluation score (see the previous section) with the score computed during the application of overlay. However, in contrast to the discourse modeler of the SMARTKOM system, FADE does not automatically perform the enrichment operation on each interpretation hypothesis. It rather has to be called explicitly via a PATE rule if required, as it is not always suitable to add all contextual information to a current interpretation hypothesis (see section 4.4.2 for a discussion).

8.7 The Long-Term Memory

As we have discussed in the previous sections, FADE's long-term memory (LTM) fulfills several tasks. One of the key tasks is of course the retrieval of knowledge about the world that has neither been mentioned during the previous discourse, nor is inferable from the physical context. To this end, the LTM of FADE can be queried by under-specified representations of objects and returns all stored information that is compatible with that request. For example, if we consider the example discourse fragment ((91) and (93) on page 208) the mode of operation of the LTM becomes clear.

After the first user turn (91), the system updates the LTM with the referring expressions that are embedded in the utterance. One of these referring expressions refers to the instance that stands for the football match Germany vs. USA at the World Cup 2002 which means that the activation of that instance is increased. The spreading activation process (see section 7.2.3) ensures that this increase in activation is passed on to associated instances. As a result, the players of the game are now accessible in the Referential Layer. During the processing of the second utterance (93), the system uses the under-specified statement *Ziege* which describes a person instance that has "Ziege" as slot-filler for the slot "lastname" in order to determine the actual referent. Since the actual instance is already activated and accessible at the Referential Layer a potential ambiguity (i. e., more than one person named "Ziege" in the knowledge base) would not cause any trouble.

8.7.1 Transferring Information from the LTM to the WM

All entities of the LTM whose activation exceeds a certain threshold (as specified in the configuration file of FADE) appear also in the Referential Layer of the discourse context.

Each entity that is not yet part of the Referential Layer appears as an Implicit Activated Object.

The spreading activation approach that transfers part of the activation of entities referenced during the discourse to associated entities ensures that only a relevant subset of the available knowledge is activated. At the beginning of an interaction the spreading activation needs to be initialized with the conversational setting (e. g., the participants, the location).

8.7.2 Updating the LTM

FADE makes direct use of the built-in LTM of PATE. After the initialization phase the LTM comprises all instances of which the system is aware. In order to update the LTM, FADE supports two approaches: (i) *question-time* and (ii) *read-time* activation spreading. Question time means that the long-term memory performs the activation spreading just when an indirect reference needs to be resolved. In contrast, read-time means that while an interaction is going on, FADE updates the LTM every time a participant of this interaction makes a contribution. Using the question-time methods has the advantage of saving CPU time as long as no indirect reference needs to be resolved, however, it has the disadvantage of slowing down the resolution process if an indirect reference occurs. The read-time generates more CPU load during the post-processing of user and system turns but has the advantage of already activated referents for indirect references. The actual processing strategy can be selected in the configuration file of FADE.

In general, updating the LTM means identifying all referring expressions mentioned in a contribution and then sequentially using these entities as a starting point for the spreading activation process. If applied after every contribution of a participant, this ensures that the activation pattern of the LTM always reflects the informational status of the interaction.

8.8 Conclusion

With FADE we have introduced a generic component for multimodal fusion and discourse processing that can be employed in a variety of multimodal dialogue systems. As we will show in the next chapter, this component can be used in quite different setups for the task of multimodal fusion and discourse processing.

FADE realizes the approach of an integrated processing of interactional and propositional actions that has been developed in this dissertation. Moreover, it provides the basis for interpreting and generating complex discourse phenomena like turn-taking behavior and backchannel feedback. A key aspect of the integrated approach of FADE is that it allows one to treat deictic, anaphoric and exophoric expressions in an integrated way. This permits one to deal with ambiguity in deictic expressions like “*this song*” which can either be a place deictic expression (if accompanied by a pointing gesture), an anaphoric reference to a previously mentioned song or an exophoric reference to a song in the physical surroundings. Usually, this ambiguity is dealt with by two distinct components. In our approach, however, the ambiguity is processed by a single component.

Another key feature of FADE is that it can easily be extended in order to be able to deal with specific requirements of new tasks or applications. As we will show in the next chapter, it is possible to use PATE rules to realize a variety of subtasks. Moreover, the reference resolution algorithms of FADE are not limited to specific surface realizations. Pointing gestures for instance, do not necessarily need to be realized by means of pen-based pointing

gestures. In fact, they also could be realized through eye gaze or head-movements if the system supports a corresponding recognizer. The only requirement is that the recognizer generates an ontological instance for such a pointing gesture that is compatible with the meta-ontology introduced in section 8.2. In the same way, FADE is able to deal with a great variety of display configurations or system turns as long as they are expressed or can be transformed into valid instances of the meta-ontology.

Chapter 9

Applying FADE in Multimodal Dialogue Systems

In the previous chapter we introduced and discussed the architecture and functionality of FADE. In this chapter we will show how FADE has been employed in three practical dialogue systems with varying requirements and capabilities. Besides showing the general operability of FADE, the purpose of this chapter is also to illustrate how FADE can be adapted to new tasks and domains. Section 9.1 gives a brief overview of the configuration files of FADE. In section 9.2 we discuss the role of FADE in VIRTUALHUMAN, a multiparty dialogue system. Section 9.3 then discusses the role of FADE in the SMARTWEB project, while section 9.4 focuses on the OMDIP project.

9.1 Introduction

The goal of this chapter is to provide a proof of concept that FADE is a generic tool for multimodal fusion and discourse processing in multimodal dialogue systems. Moreover, it also shows what needs to be done in order to adapt FADE to a new system or application. To this end, we will discuss three distinct dialogue systems that employ the core FADE engine for the task of multimodal fusion and reference resolution. We will show in particular which parts of these tasks are already covered by the built-in generic functionality of FADE, and which parts need to be realized by means of specific PATE rules.

Before we go into any detail, we will briefly discuss what needs to be done to integrate FADE into a dialogue system or application. First, we will show how FADE is initialized by using public API methods. Then, we will introduce the central configuration file of FADE before we show how the data exchange between FADE and a host system is realized.

9.1.1 Initializing the System

FADE is initialized via the constructors of two Java-classes: *PateMain(String path)* and *FadeMain(String path)*. Both constructors expect a pointer to the central configuration file of FADE (see the next subsection) in order to perform the necessary initialization (e. g., loading the type-system, loading PATE rules).

9.1.2 The Central Configuration File

The central configuration file of FADE is organized by means of namespaces reflecting different information groups that can be used to adjust individual aspects of the system. Examples of namespaces and entries that need to be specified are:

productionSystem

- *rules*—the path and file names of the rule files for PATE.
- *typeHierarchy*—the path and file-name where the type hierarchy or ontology is located.
- *pluginDirectory*—specifying the directory where the PATE plug-ins are located.
- [...] (e.g., a number of parameters for controlling PATE’s rule weighting and spreading activation mechanism)

dim

- *pathToInterpretation*—defines the location of the interpretation hypotheses within the interpretation lattice.
- *pathToFragments*—defines the location of the interpretation of elliptical and fragmental expression within the interpretation lattice.
- [...]

9.1.3 Importing the Type System

FADE’s importer for the type system and the knowledge base currently supports five different formats:

- PATE’s internal type system;
- The internal *Oiled* format;
- The internal *Protégé* format;
- The RDF/S format;
- The OWL/RDF format.

Any ontology that is used for FADE must either incorporate the meta-ontology defined in section 8.2, or must directly comprise the specified classes and properties.

9.1.4 Registering Communication Channels

The communication between FADE and other system components is realized via a central input channel, the *perception module* (see section 8.4). All incoming data is pre-processed by this component and then routed either to PATE or the discourse context.

On the output side, FADE provides the possibility to specify classes as output channels that can be accessed within PATE. This is realized by implementing the Interface *OutputQueue*.

9.1.5 Initializing the Immediate Conversational Context

The immediate conversational context of FADE needs to be initialized before the system can be used. As discussed in Chapter 8, the conversational context is not a fixed data structure but encompasses a representation of the conversational situation that is tailored to the system or application at hand. A minimum requirement, however, is that the conversational context comprises descriptions of the individual participants (i. e., the user(s) and the system or a number of virtual characters).

Besides this, the conversational context can encompass all kinds of additional information needed for the tasks at hand. The only restriction is that the information needs to be compatible with the ontology. In SMARTWEB for example, we represent the *AttentionalStatus* (see section 9.3.3).

9.2 FADE in the VirtualHuman System

The VIRTUALHUMAN system is the first of the three systems into which FADE has been integrated. Compared to the other two systems (see below), VIRTUALHUMAN poses somewhat different requirements to FADE as it is a multiparty dialogue system where three virtual characters engage in an interaction with two human participants.

Since the project and the system were briefly introduced in section 1.4, here we will only focus on the technical aspects of the system and the role of FADE within the system. However, to provide a better understanding of the capabilities of the system, we will start with a brief description of the scenario of VIRTUALHUMAN and the tasks of FADE.



Figure 9.1: Screen-shot of the VIRTUALHUMAN system during the first game phase with three virtual characters present. From left to right: The moderator, the virtual football expert Mr. Kaiser and the virtual football expert Ms. Herzog.

9.2.1 The Scenario of VirtualHuman

The basic scenario of the final VIRTUALHUMAN demonstrator is a football quiz show where two human contestants interact with three virtual characters: A moderator and two virtual experts named Ms. Herzog and Mr. Kaiser. The quiz consists of two phases; in the first phase the moderator shows short videos of suspenseful situations of past football games.

These videos stop just when the situation is about to be resolved (e. g., the striker attempts to shoot) and the human contestants have to guess how the situation will go on. In order to ease the task for the human contestants, the moderator provides three possible answers and the human contestants also have the opportunity to ask the virtual experts for their opinion. After three rounds of videos, the moderator selects the winner of this first phase (figure 9.1 depicts a screen-shot of the first phase of the quiz).

In the second phase, the winner of the first phase will be given the opportunity to assemble the line-up of the German national football team. The scenery of the game changes to the one shown in figure 9.2. The moderator and the female expert are positioned behind a 3D representation of a football field with marked positions for the players while the available football players are displayed in a list on the right side of the screen. The human user can place players either by means of spoken or multimodal commands (see the sample dialogue in the next section).

The game ends either after a predetermined time interval or when the user has assembled a complete football team. In any case, the moderator and the virtual expert will evaluate the assembled team and discuss the positive and negative aspects of the team.



Figure 9.2: The studio during game stage two with two virtual characters. The lower part of the picture shows the football field where the line-up is visualized.

An Example Interaction

Since a typical interaction with the VIRTUALHUMAN system lasts about 15 minutes, we will give here only two short fragments of the game stages. For the first stage, we will start just

when the moderator shows the first video sequence¹:

Moderator: [gazes at User 1 and User 2] “*Ok, here comes the first video.*” [The video is shown on the projection screen behind the virtual characters]

Moderator: [gazes at User 1 and User 2] “*What happens next: first [counting gesture] the goalkeeper will save the ball; second [counting gesture], the striker will score; third [counting gesture], the striker will miss the goal.*”

Moderator: [gazes at Ms. Herzog] “*Ms. Herzog, what do you think?*”

Herzog: [gazes at the moderator] “*Well, I think the goalkeeper saves the ball.*”

Moderator: [gazes at User 1] “*Ok, User 1, what do you think?*”

User 1: “*Hhm, I think Ms. Herzog is right.*”

Moderator: [gazes at User 1 and then at User 2] “*Well, I don’t know, User 2, what do you think?*”

User 2: “*I guess the striker scores.*”

Then the moderator shows the complete video sequence and awards the points (a correct answer is worth one point). After three rounds of videos, the winner of the first stage (i. e., the human player with the highest score) will proceed to the second stage.

After a brief introduction to the line-up game, the moderator starts the second stage:

Moderator: [gazes at User 1] “*Ok, let’s get started.*”

User 1: “*Put Oliver Kahn into the goal.*”

Herzog: [nods; gazes at User 1] “*That’s an excellent move!*”

Moderator: [gazes at Ms. Herzog; nods; gazes at User 1] “*Great, Kahn in the goal position.*”

User 1: “*Ms. Herzog, give me a hint!*”

Herzog: [smiles; gazes at User 1] “*I would definitely put Ballack into the central mid-field.*”

User 1: “*Ok, let’s do that.*”

Herzog: [smiles; nods; gazes at User 1] “*You won’t regret this move.*”

Moderator: [nods] “*Great, Ballack as central midfielder.*”

User 1: ... [hesitates]

Moderator: [gazes at User 1; encouraging gesture] “*Don’t be shy!*”

User 1: “*Hhm, put Metzelder to the left of Ballack.*”

¹Note that the contributions of the virtual characters and the users were translated from the German original.

This interaction goes on until the user has assembled a complete team and indicates that he is finished, or until the user runs out of time. At the end of the game the moderator and the virtual expert evaluate the team and discuss potential problems for the individual parts of the team.

9.2.2 Tasks of FADE in VirtualHuman

The individual tasks of FADE in the VIRTUALHUMAN system are:

Reference resolution

Resolution of spatial references

(103) **User:** “*Ballack right of Frings.*”

(104) **User:** “*Ballack in front of Lehmann.*”

For the resolution of spatial references, FADE needs to maintain a representation of the spatial organization of the football field displayed in the virtual studio. The actual reference resolution is then realized by FADE’s built-in methods for reference resolution.

Resolution of discourse references

(105) **Herzog:** “*I think Ballack shoots the ball into the evening sky.*”

Kaiser: “*I am sure Ballack scores.*”

Moderator: “*Player 1, what’s your opinion?*”

User: “*I think Ms. Herzog is right.*”

The recognition result of the user’s last utterance is the dialogue act *Agree* comprising the information that the user agrees with the character Ms. Herzog. However, this dialogue act still lacks the semantic content of what Ms. Herzog actually said and thus the task of FADE—that represents the moderator—is to resolve what has actually been said by Ms. Herzog before the action planner is able to process the *Agree*.

(106) **Herzog:** “*I would place Ballack into the central midfield.*”

User: “*Ok, let’s do that.*”

This example basically requires the same functionality of FADE. The user reacts to the *Propose* dialogue act of Ms. Herzog with a *Confirm* which again needs to be enriched with the semantic content of Ms. Herzog’s original utterance. This is realized via FADE’s built-in methods for resolving discourse deixis.

Resolution of deictic references

(107) **User:** “*This player [pointing gesture] into the midfield.*”

In this example, the user selects the player by using a deictic reference and an accompanying pointing gesture—a classic fusion task. Again, this functionality is realized by using FADE’s built-in methods for reference resolution.

Addressee identification

(108) **User:** “*Ms. Herzog, what do you think?*”

In this case, the identification of the intended addressee is rather easy since the user has already provided the addressee through the vocative *Ms. Herzog*. However, the next example is more complex:

(109) **Moderator:** “*Tell me when you are ready.*”
User: “*I am ready.*”

In this example, the intended addressee can be inferred through the adjacency pair that is formed by the *Request* and *Confirm* dialog act. Thus, the addressee would be the previous speaker. However, if no information is present to narrow down the intended addressee, each character assumes they have been addressed. Then the characters have to decide based on their capabilities and current aims whether they will react or not. Consider, for example, the following utterance by the user in the second game stage:

(110) **User:** “*Put Michael Ballack into midfield.*”

From the perspective of FADE it is not clear in this case whether the moderator or Ms. Herzog has been addressed. Consequently, the individual FADE components of the two characters assume to be the ones addressed. However, only the moderator’s dialogue manager (see below) is able to deal with the user utterance, and Ms. Herzog’s action planner would ignore it. As outlined in section 8.5.2, this functionality cannot be realized by means of generic built-in methods. Section 9.2.5 shows how the identification of the intended addressees is realized.

Triggering backchannel feedback

Another important task for FADE in the VIRTUALHUMAN system is to trigger the gazing behavior of the individual characters. If, for example, someone starts speaking, it is natural for the listeners to react by gazing at the speaker in order to indicate their attention and their general acceptance of this participant as the speaker (see section 3.3.2). Moreover, it is also important for the speaker to look at the intended addressee(s) at the end of the turn in order to encourage them to take over the turn. Section 9.2.5 shows how this is realized by means of PATE rules.

9.2.3 The Architecture of the VirtualHuman System

The VIRTUALHUMAN demonstration system consists of three standard PCs (one for speech recognition² (ASR), one for 3D-rendering (graphic output), and one for the dialogue) and a 3D presentation system (consisting of two high-resolution video projectors). The two human users stand in front of two columns, each equipped with a microphone (Sennheiser) and a track-ball controlling a mouse pointer.

The basic architecture of the VIRTUALHUMAN system is partitioned into four main function blocks (see figure 9.3):

²Since there are two users, two ASR engines run on this computer.

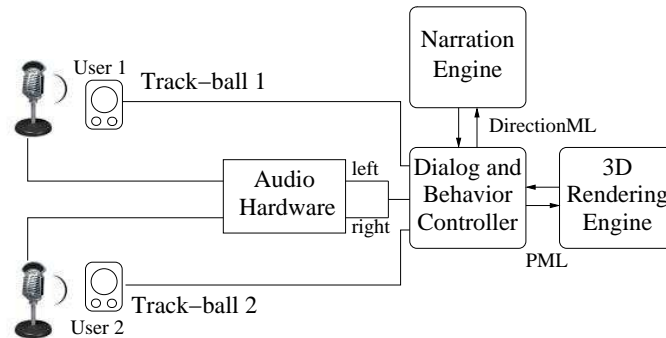


Figure 9.3: Basic Architecture of the VIRTUALHUMAN system.

Input Devices—The input devices of VIRTUALHUMAN comprise two microphones (one for each human user) and two track-balls. The output of the two microphones is first routed to a standard audio mixer where the two signals are assigned to different audio channels (left and right). Then, the signals are routed to the Behringer Autocom device which is a 2-Channel Expander/Gate/Compressor/Peak Limiter with Integrated Dynamic Enhancer, De-Esser and Low Contour Filter. This device is usually employed for high-quality audio recordings in studios or on stage. However, we have found that this device can also help conducting open-microphone speech recognition in noisy surroundings. It can be used to suppress all background noise below a certain threshold, and to limit peaks in the audio signal without corrupting the signal. Especially the suppressed background noise drastically improves the performance of the ASRs.

NarrationEngine—The NarrationEngine (NE) of VIRTUALHUMAN monitors and controls the development of the interaction between users and characters. To this end, it can prescribe *goals* for the individual virtual characters. These aims trigger particular behavior and action sequences of the characters. Consider, for example, a goal like *greet* which causes a character to greet the character that is stated in the body of the goal. The NE also receives feedback from the individual characters whether they were able to fulfill the prescribed aims or not. The basic approach of the NarrationEngine is described in Göbel et al. (2006).

CDE Controller The dialogue and behavior controller of VIRTUALHUMAN comprises a set of *Conversational Dialogue Engines* (CDEs; see Löckelt and Pflieger (2006); Löckelt and Pflieger (2005); Pflieger and Löckelt (2005)) which represent the individual participants of the scene. The dialogue controller takes the aims sent by the NE and distributes them to the individual CDEs (the language used for this information exchange is called *DirectionML*). We will discuss the CDEs in detail later in this section.

3D Rendering Engine—The 3D rendering engine of VIRTUALHUMAN is called *Avalon*³. Avalon is a high-performance 3D rendering engine that supports a variety of displays ranging from simple Web browsers to complex projection grids. The dialogue engine communicates with the Avalon player via PML (*Player Markup Language*) scripts which encode precisely timed sequences of verbal and nonverbal actions.

³See <http://www.zgdv.de/avalon/>.

9.2.4 The CDE Framework

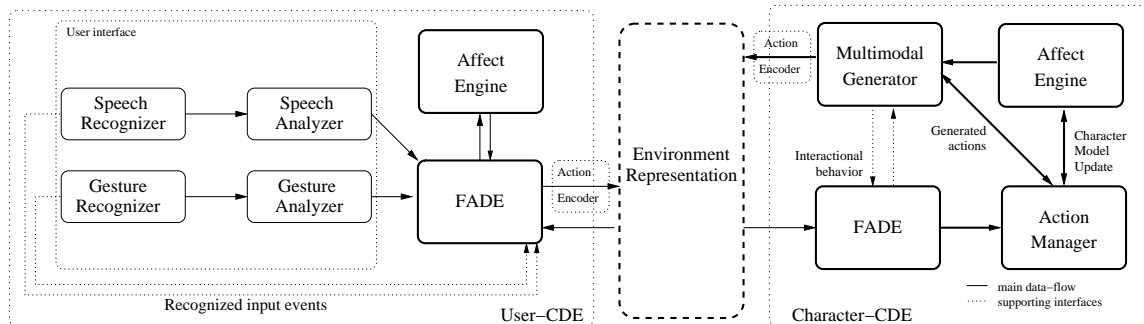


Figure 9.4: Architecture of the User-CDE and Character-CDEs of the VIRTUALHUMAN system.

On the top-level of the CDE framework there is the *CDE controller* which controls and monitors the individual actions of the CDEs that participate in the interaction. Each participant of the interaction (virtual or human) is represented by its own CDE. Consequently, there are two different types of CDEs (see figure 9.4): (i) User-CDEs and (ii) Character-CDEs which we will introduce in detail in section 9.2.4. But before that we will briefly discuss the internal knowledge representation of VIRTUALHUMAN.

Knowledge Representation in VirtualHuman

The knowledge representation of VIRTUALHUMAN is based on an ontology that was developed from scratch for this project. The upper-level is based on the one suggested in (Russell and Norvig, 1995). For the purpose of inter-CDE communication, we added a set of abstract classes like, for example, dialogue acts and nonverbal acts.

The *Act* class marks the top-level of a hierarchy of four kinds of acts: *PhysicalActs*, *DialogActs*, *NonverbalActs* and *StartOfSpeech* (see figure 9.5 for an excerpt of the dialogue acts of VIRTUALHUMAN). A fully instantiated dialogue act comprises three key slots: (i) the speaker of that utterance (*has_initiator*), (ii) the addressee (*has_addressee*) and (iii) the semantic content of the utterance (*has_content*). The ontology of VIRTUALHUMAN furthermore incorporates the complete meta-ontology of FADE as described in section 8.2.

User CDEs

A User-CDE represents a user of the VIRTUALHUMAN system. The task of a User-CDE is to convert the recognized verbal and nonverbal actions by the user into instances of the ontology that can be processed by the other CDEs. To this end, a User-CDE consists of a set of components: (i) a speech recognizer, (ii) a gesture recognizer, (iii) a natural language understanding (NLU) component and (iv) an instance of FADE.

The Speech Recognizer The speech recognizer used for VIRTUALHUMAN is the *ISIP* speech recognizer developed by the *Institute for Signal and Information Processing* at Missis-

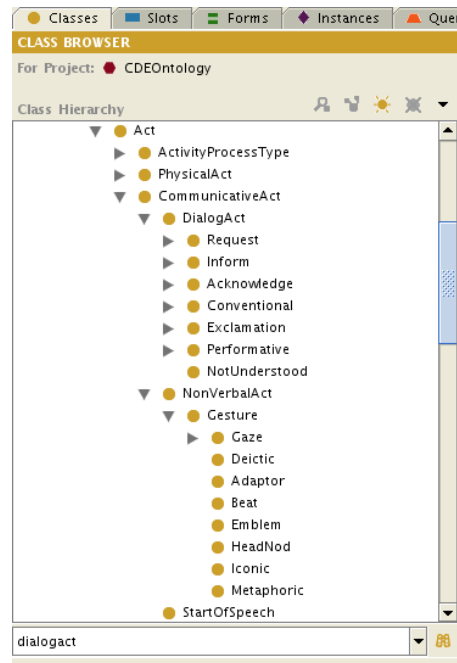


Figure 9.5: Overview of the Act hierarchy of VIRTUALHUMAN.

sippi State University. This speech recognizer is freely available⁴ and supports on-line updates of the language models. Each User-CDE has its own speech recognizer that can be located on a different machine.

All recognized speech input is routed to the natural language understanding component of a User-CDE. A *start-of-speech* event is immediately routed to the CDE controller which in turn forwards it to all CDEs.

The Gesture Recognizer/Analyzer The task of the gesture recognizer/analyzer of the VIRTUALHUMAN system is to map the recognized pointing gestures to instances of the ontology. Typically, a pointing gesture in VIRTUALHUMAN either selects an entry of a multiple-choice menu presented on the screen, or an object that is located in the virtual world (e. g., a box representing a football-player, or a position on the football-field). Moreover, there are two types of selected objects: (i) those that can be interpreted on their own (e. g., selected menu items) and (ii) those that make only sense when interpreted in combination with accompanying speech input or additional pointing gestures (selecting a football-player and subsequent positioning).

The Natural Language Understanding Component (NLU) The natural language understanding component (NLU) of VIRTUALHUMAN interprets the recognized spoken input of the user and converts it into instances of the ontology. This component is realized by means of PATE. The output of the speech recognizer is split into the individual words which in turn

⁴See <http://http://www.cavs.msstate.edu/hse/ies/projects/speech/index.html>.

are pushed onto the goal-stack of the NLU component. The PATE rule `interpretWords` subsequently looks-up the words in the LTM and adds the resolved words (that now comprise a semantic meaning) to the working memory. When all words have been pre-processed, the actual interpretation process takes place where a set of 70 rules is applied to the working memory in order to create a coherent representation of the user's utterance.

Character-CDEs

A Character-CDE represents an autonomous virtual character that is part of the virtual world. The architecture of a Character-CDE comprises four main components (see figure 9.4): (i) an instance of FADE which is responsible for multimodal fusion, discourse processing and reactive behavior, (ii) an action manager that is responsible for the deliberative actions of the character, (iii) an affect engine which maintains the affective state of the virtual character and (iv) a multimodal generator that generates synchronized verbal and nonverbal output of the character. In the following, we will briefly describe the last three components.

The Action Manager The task of the action manager of a Character-CDE is to plan, execute, and monitor the deliberative actions of the character. What kind of actions the action manager triggers, depends on the goals set by the narration engine, the actions of the other participants (both virtual and human) and the affective and internal state of the character. A detailed description of VIRTUALHUMAN's action management is given in (Löckelt, 2005, 2007) here we will only give a brief overview of its basic functions.

The processing model of VIRTUALHUMAN's action manager is based on the dialogue games approach (e. g., Mann (2002)). This approach views interactions in terms of rule-based exchanges of dialogue acts where the rules define legal sequences of dialogue acts. Thus, the rules help the participants of an interaction to synchronize their actions so that they can reach a joint goal. That way, the rules reflect social conventions accepted by all participants. A question, for example, triggers the expectation that the addressee of that question will either attempt to answer it or provide some kind of feedback. Formally, a dialogue game consists of preconditions and postconditions that are applied to the current internal goal state by the action manager in order to determine valid moves that help to fulfill the goals. Thus, the primary trigger for actions are the goals that can either be set by the Narration Engine or by changes in the internal state of the character (e. g., a strong affect can cause the character to complain).

The Affect Engine The affect engine of VIRTUALHUMAN maintains the affective state and controls the idle behavior of the characters. It is based on a computational model of affect incorporating three different types of affect (Gebhard, 2005):

- 1 Emotions** are considered short-term affect that decay after a rather short period of time. Emotions influence facial expressions, facial complexions (e. g., blushing), and the performance of conversational gestures.
- 2 Moods** are considered medium-term affects that are not related to a concrete event. Moods are longer lasting affective states with a great influence on the cognitive functions of humans.

3 Personality is considered a long-term affect reflecting individual differences in mental characteristics.

In VIRTUALHUMAN, the action manager annotates the perceived actions of the individual characters with appraisal tags and dialogue acts and sends this information to the Affect Engine. The output of the Affect Engine consists of an updated affective state of the character which influences its further behavior.

The Multimodal Generator The task of the multimodal generator is to transfer an abstract ontological instance of a dialogue act into a multimodal contribution represented in PML. The output of the generator contains the spoken utterance as well as tightly synchronized nonverbal actions (gazes, adaptors, emblematic, iconic, deictic and beat gestures). See (Kempe et al., 2005) for more information about VIRTUALHUMAN's multimodal generator.

The generator is a compound module of four different parts (see (Kempe, 2005)): The main control is performed by an instance of PATE. Upon receiving a dialogue act from the action planner, production rules manage initial turn-taking and pensive behavior. PATE also restructures dialogue acts and annotates them with grounding information from the conversational context provided by FADE. The content of a dialogue act is preprocessed by NIPSGEN a production rule system based on the SPIN system (see Engel (2002, 2006)) which uses generic rules to convert the complex dialogue act into operations on elementary syntax trees. These syntax trees are then turned into a derivation tree by the LTAG generator (see Becker (2006)). The resulting derivation tree is split into different clauses and words. The sentence is sent to the text-to-speech (TTS) engine and the word boundaries of its verbalized output are analyzed. This enables the PATE-based gesture rules to select suitable gestures and to exactly time them at word boundaries. In the last processing step, PATE converts the gesture instances and the sentence into a PML document and sends the PML code to the CDE controller.

9.2.5 FADE's Processing Strategies for VirtualHuman

In this section we will discuss the processing strategies of FADE that differ from the generic strategies discussed in chapter 8. Since the reference resolution mechanism makes use of the built-in methods for reference resolution it is not discussed here.

Updating the Conversational Context

The conversational context of VIRTUALHUMAN encompasses information about the present participants (both human and virtual characters) and the physical surroundings (the studio and in the second stage of the quiz show the football field where the players are placed). Furthermore, the conversational context comprises a slot called *mySelf* where the ontological description of the character it stands for is represented.

The conversational context is updated using PATE rules. There is, for example, a PATE rule that incorporates new participants into the context structure. Another set of rules deals with updating the physical surroundings and here in particular with updates of the football field. If a new player is added to the football field or a previously placed player is removed, specialized rules ensure that the physical environment of the conversational context will be updated accordingly.

Besides changes in the physical surroundings, the conversational context is also updated every time a participant starts or finishes to speak. A set of PATE rules deals with the respective dialogue acts and updates information about the current and previous speaker and addressees. If a virtual character starts to perform an interactional nonverbal behavior, this is stored in the conversational context.

Updating the Discourse Context

The discourse context of a virtual character is updated every time FADE finishes the analysis phase and sends a dialogue act to the action manager. Each dialogue act is first of all incorporated into the turn sequence (thereby including information about the speaker of that dialogue act and the addressees). In a second step, all referring expressions are incorporated into the Referential Layer of the discourse model.

If a participant performs a nonverbal action that is classified as a propositional contribution, this action is also incorporated into the discourse context, i. e., a Nonverbal Action is added to the Modality Layer and a linkage to the referenced object at the Referential Layer is realized.

Addressee Identification

As outlined in section 8.5.2, the addressee identification is realized by means of PATE rules. In this section we also discussed a hierarchical set of restrictions that can be used to determine the intended addressees of a contribution. Because of some constraints of the VIRTUALHUMAN system, not all of these restrictions could be employed for the actual system. It was, for example, not possible to use the gaze restrictions since the VIRTUALHUMAN system did not provide any means to detect the gazing direction of the users. The same holds for the body orientation and the pointing gestures (i. e., the user only has the possibility to use the mouse as a pointing device which totally inhibits pointing gestures at the virtual characters). However, if one or more of these features were available, it would be no problem to implement the respective PATE rules. Nevertheless, the missing restrictions can be compensated by the structure of the quiz show. Due to his status as the host of the show, the moderator is in most cases implicitly addressed.

The following enumeration describes the individual rules that realize the identification of the intended addressees in the VIRTUALHUMAN system:

Vocative: If the utterance of the speaker contains a vocative, the participants that are denoted by the vocative are the addressees of that utterance.

Speaker mentions participant: If the speaker explicitly names a participant within the utterance and not at a vocative position, this participant is most likely not the intended addressee.

Contextual factors 1: If none of the previous rules can be applied, the previous addressee is the current speaker and the dialogue act of the previous utterance is a subtype of Request, then take the previous speaker as addressee.

Contextual factors 2: If none of the previous rules can be applied and one of the previous addressees is the current speaker, then take the previous speaker as addressee.

```

<rule name="identifyAddressee(ContextualFactors2)">
  <weighting> 0.3 </weighting>
  <conditions>
    <condition name="goal">
      <object type="DialogAct">
        <slot name="has_addressee">
          <empty/>
        </slot>
        <slot name="has_initiator">
          <variable name="speaker"/>
        </slot>
      </object>
    </condition>
    <condition name="retrieveLastSpeaker" method="restricted">
      <object type="ConversationalStatus">
        <slot name="has_lastSpeaker">
          <object type="Character"/>
          <variable name="lastSpeaker"/>
        </slot>
        <slot name="has_lastAddressee">
          <object type="Character"/>
          <variable name="speaker"/>
        </slot>
      </object>
    </condition>
  </conditions>
  <actions>
    <action type="updateWME" name="goal">
      <object type="DialogAct">
        <slot name="has_addressee">
          <variable name="lastSpeaker"/>
        </slot>
      </object>
    </action>
  </actions>
</rule>

```

Figure 9.6: Example rule for identifying the intended addressee in the VIRTUALHUMAN system.

If no clear addressee could be identified, then consider all active, ratified participants as being addressed. Figure 9.6 shows an example rule realizing the **contextual factors 2** restriction.

Triggering Reactive Feedback

As discussed in Chapter 3, participants of an interaction make use of a variety of reactive behaviors that are meant to provide feedback for the other participants. The virtual characters of the VIRTUALHUMAN system are capable of displaying at least parts of these behaviors. An example would be the turn-taking behavior that is displayed by the virtual characters.

Before any reactive behavior can be displayed, two processing steps need to be taken: First, an appropriate communicative function needs to be selected. In a second step, the most suitable communicative behavior that realizes the communicative function needs to be selected. Which reactive communicative function is most suitable is either selected by FADE (during the interpretation phase of perceived monomodal events) or by the multimodal generator (during the planning of contributions). In the first case, FADE decides on the basis of the current conversational context when it is appropriate to display turn-taking behavior.

Communicative Function	Communicative Behavior
Give Turn	Gaze at turn-requester.
Want Turn	Raise hands into gesture space; start gesticulating.
Take Turn	Glance away, start talking.
Accept Speaker	Look at speaker, nod head.

Table 9.1: Examples of communicative functions and their behavioral realizations in VIRTUALHUMAN.

In the second case, the multimodal generator retrieves the current conversational status via FADE. If the floor is not available, the multimodal generator starts triggering nonverbal behavior that signals its wish to take the turn to the other participants (see figure 9.7).



Figure 9.7: Example of an interrupting gesture: The female character Ms. Herzog wants to take the turn and signals this to the male character—who is currently holding the turn—by raising her right arm into the gesture space.

FADE is responsible for the reactive part of these behaviors and triggers an appropriate communicative behavior when given a communicative function. Table 9.1, summarizes the mappings between communicative function and communicative behavior. An example of the communicative function *Want Turn* is given in figure 9.7. Here, the female character Ms. Herzog signals that she wants to take turn while the male character is still holding it.

9.3 FADE in the SmartWeb System

The second system FADE has been integrated into is the SMARTWEB system (see Wahlster (2004b)). As outlined in section 1.4, SMARTWEB is a multimodal dialogue system that differs from the VIRTUALHUMAN system in several aspects. First of all, SMARTWEB supports only dyadic interactions and there is no virtual character or talking head displayed on the screen. Moreover, due to the question-answering approach, the interaction patterns differ strongly

from those of the VIRTUALHUMAN system.

In the following sections we will give a brief overview of the architecture and the basic scenario of SMARTWEB before we will describe the tasks and processing strategies of FADE.

9.3.1 The Architecture of the SmartWeb System

The architecture of SMARTWEB consists of three distinct processing layers (see figure 9.8): (i) the mobile input/output processing layer (running on a PDA), (ii) the remote dialogue server (Reithinger et al., 2005a) and (iii) the knowledge server providing access to the semantic knowledge bases, the question-answering subsystem, the Semantic Composer providing access to Web services, and the Web Crawlers. In what follows we will focus on the dialogue server since this is the component that hosts the dialogue system and FADE. However, we will also discuss the Semantic Composer since this component is closely related to the dialogue manager.

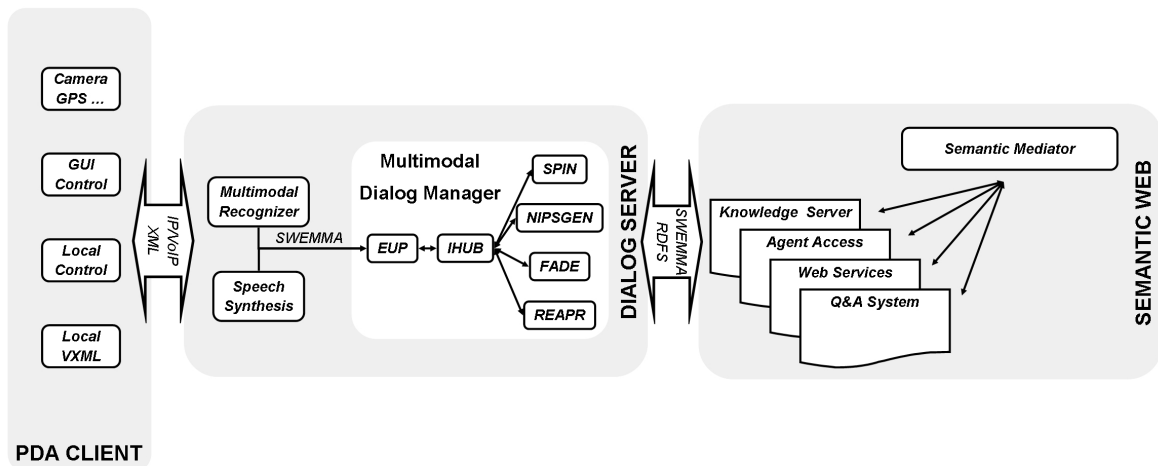


Figure 9.8: The basic architecture of the SMARTWEB system.

Besides FADE the dialogue manager of SMARTWEB comprises five core components that are integrated into the *IHub* framework (Reithinger and Sonntag, 2005). The *IHub* provides a *hub-and-spoke* based system architecture that connects various components of a dialogue system. In the following we will give a brief description of the remaining five components.

Natural Language Understanding The natural language understanding component of SMARTWEB is called SPIN (Engel, 2006, 2005). SPIN is a free-word-order parser that directly transforms recognized input into ontological instances. An important aspect of SPIN is its template language that supports easy creation and maintenance of the knowledge base as well as fast and robust processing.

SitCom The SitCom component receives the integrated multimodal user request from FADE. Its task is to enhance the semantic representation of the user request with contextual

information from the situation (see Porzel et al. (2006)). This information could be based on the current weather conditions, the current location, the current time, . SitCom then sends the enriched user query to the REAPR component.

Reaction and Presentation Planning The reaction and presentation planning component (REAPR; see Sonntag (2006)) is responsible for the basic dialogue management and the high-level planning of the system presentations for the user. REAPR is based on a finite-state-automaton that operates on an information space (IS). REAPR receives input from SPIN (the selected word sequence and a paraphrase representing what the system understood) and FADE (the semantic description of the selected interpretation hypothesis). These intermediate results are then presented on the PDA screen and can be edited by the user. If the user changes the paraphrase or original input, the message is sent back to the originating module for reinterpretation. Finally, REAPR sends the user's request to the Semantic Mediator. The results obtained from the Semantic Mediator are prepared for presentation and rendered on the mobile device.

Text Generator The text generation is realized by NIPSGEN which uses the same parser as the natural language understanding component in combination with a TAG based grammar. The input of this component are ontological instances that represent search results. These results have to be transferred into textual representations that are displayed on the screen and a short answer that is synthesized for the user. The actual transformation of the ontological structures into text is organized in four processing steps:

- 1 An intermediate representation is built on a phrase level.
- 2 The intermediate representation is converted into a derivation tree for the TAG grammar.
- 3 The syntax tree is constructed and the features of the tree nodes are unified.
- 4 The correct inflections for the lexical leaves are looked up in the lexicon before the eventual result is generated for the lexical leaves.

Semantic Composer The Semantic Composer is the interface between the dialogue system and specialized Web services. The task of the Semantic Composer is first to interpret the ontological user queries and then to plan and execute an appropriate sequence of Web service queries in order to resolve the request. A key feature of the Service Composer is that it is able to detect under-specified user queries and to trigger clarification questions. This request for a clarification question is also processed by FADE since subsequent user contributions are likely to be elliptical answers the question posed by the system. See Pfalzgraf (2006) for a detailed description of the Semantic Composer.

The Scenario

The general idea of SMARTWEB is to provide a generic and intuitive multimodal interface to structured knowledge sources. The ultimate goal of SMARTWEB is to assist the user during information retrieval in an intelligent way. To this end, the SMARTWEB user interface is realized by means of a multimodal context-aware and situated dialogue system. Thus, the



Figure 9.9: SMARTWEB in action: The screen-shot on the left depicts the summary of the user's question, the screen-shot in the middle depicts the system response and the screen-shot on the right depicts a detailed view of an answer.

user can query and access the different knowledge sources via a single interface. Figure 9.9 shows three different views of the SMARTWEB system: (i) the paraphrase which provides feedback about what the system understood, (ii) the presentation of a set of results and (iii) a detailed view of a single result.

The following sample interaction exemplifies typical questions and interaction patterns SMARTWEB is able to deal with:

- (111) **User:** *"When was Germany world champion?"*
- (112) **System:** *"In the following 4 years: 1954 (in Switzerland), 1974 (in Germany), 1990 (in Italy)."*
- (113) **User:** *"And Brazil?"*
- (114) **System:** *"In the following 5 years: 1958 (in Sweden), 1962 (in Chile), 1970 (in Mexico), 1994 (in USA), 2002 (in Japan)."* [displays picture of the line-up in 2002]
- (115) **User:** [Pointing gesture at player Ronaldo] *"How many goals did this player score?"* (see figure 9.10)
- (116) **System:** *"15 times."*
- (117) **User:** *"What can I do in my spare time on Saturday?"*
- (118) **System:** *"Where?"*
- (119) **User:** *"In Berlin."*
- (120) **System:** *"The cinema program, festivals and concerts in Berlin are displayed on the screen."*



Figure 9.10: Example interaction with the SMARTWEB system: The user points at a player displayed on the screen while uttering: “How many goals did this player score?”.

9.3.2 Tasks of FADE in SmartWeb

FADE fulfills the following tasks within the SMARTWEB system:

Resolution of elliptical expressions

(121) **User:** “Who was world champion in 1990?”

System: “Germany.”

User: “And in 2002?”

(122) **User:** “I want to travel from Saarbrücken to...” [The system failed to catch the last word]

System: “Where do you want to go?”

User: “Berlin.”

These types of elliptical expressions can be resolved by means of the built-in methods provided by FADE’s API without any modification.

Reference resolution

Deictic references

- (123) **User:** [Points at a picture of a football team] *“How often has this team been the world champion?”*

Anaphoric references

- (124) **User:** *“Who was world champion in 1990?”*
System: *“Germany.”*
User: *“How often has this team been the world champion?”*

Spatial references

- (125) **User:** *“Who is the player right of Ronaldo?”*
(126) **User:** *“Who is the third from the top right?”*

Temporal references

- (127) **User:** *“What is the weather going to be like tomorrow?”*
(128) **User:** *“What is the weather going to be like in three days?”*
(129) **User:** *“What movies are playing on Saturday?”*

All these types of referring expressions can be processed by FADE without any modification of the reference resolution mechanism. The only aspect that requires some adaptation work is the update of the physical environment since a different representation format is used for the displayed objects. This is realized by means of a plug-in called “UpdatePhysicalContext” which converts the MPEG7 based representation into FADE’s internal format for representing the spatial organization of physical environments.

Disambiguation of interpretation hypotheses

- (130) **User:** *“Give me more information about this.”* [accompanying pointing gesture]
Here, the task of FADE is to select the most suitable interpretation for that utterance since SPIN generates multiple hypotheses for the object representing *about this*.

Fusion of OnView and OnTalk messages

SMARTWEB constantly monitors the attentional status of the user by means of an OnView and an OnTalk recognizer. These two recognizers provide independent measures of whether the user is addressing the system or not. The task of FADE is to combine these two measures in a coherent representation of the attentional state of the user (see the next section).

9.3.3 Processing Strategies of FADE in SmartWeb

In this section we discuss specific processing strategies of FADE in the SMARTWEB system. As with the VIRTUALHUMAN system, most of the generic methods for dealing with elliptical and referring expressions can be used without modifications. However, there are some aspects where specific PATE rules and plug-ins are needed. Appendix A provides a processing trace of a FADE for about ten user utterances.

Updating the Context Models

Incorporating User and System Turns The eventual semantic representation of a user turn is sent to FADE by the SitCom module. It is incorporated into the discourse context model by using the standard method of FADE's API for integrating contributions of participants. As outlined in section 8.6.4, this update is done in three steps: (i) first all referring expressions contained in the contribution are identified, (ii) each referring expression is integrated into the discourse context and (iii) the activation values of all Discourse Objects are updated eventually.

System turns are sent to FADE by the REAPR component. Since SMARTWEB's main purpose is to serve as a question-answering system, system turns are either a response to a previous question or a clarification question. In case of a response, FADE identifies the main answer in the answer table and updates the context accordingly. If, for example, the user asked for the winner of the World Cup in 1990, the main answer would be the semantic representation of the German World Cup team in 1990. The system turn presented to the user then is an elliptical verbal message conveying the main answer (in this case "*Germany*") that is also displayed on the screen. This means that FADE has to incorporate the main answer as the focused Discourse Object into the context model.

If the system turn is on the other hand a clarification question, FADE identifies the focus of the question and updates the context model accordingly. This focused object then serves as an anchor for the integration of an elliptical user response (see example (122)). Besides user and system turns, FADE also receives updates when the content of the display changes. Such a display update message contains a semantic representation of the currently displayed object(s). Since these updates are rather system specific, they are incorporated into the context representation using specialized PATE rules (see below).

Display Updates Every time REAPR sends an update of the graphical display to the client, FADE also receives the updated display representation. This display representation is represented by means of instances of the SmartMedia ontology (see section 5.4.2). However, since FADE uses another format for representing spatial relations, the original display update needs to be converted into the internal format. This is done by means of a PATE plug-in.

Consider as an example a SmartMedia instance representing a group-picture of a football team (as in figure 9.11). Each player in such a picture is represented by means of an instance of the class *StillRegion*. Such an instance comprises information about the player and the coordinates of the bounding box surrounding the player in the picture. Given this information, FADE has to deduce the spatial organization of the picture (e. g., the rows and the order of the players).

Resolving Spatial References

In SMARTWEB the user can select objects and persons by means of spatial references. Consider, for example, the image depicted in figure 9.11 which shows the lineup of the Brazilian football team of 1998. In this example the user has the possibility to refer to the individual players by means of spatial references like:

(131) **User:** "*Who is the fourth player from the lower left?*"

(132) **User:** "*Who is the player right of Ronaldo?*"

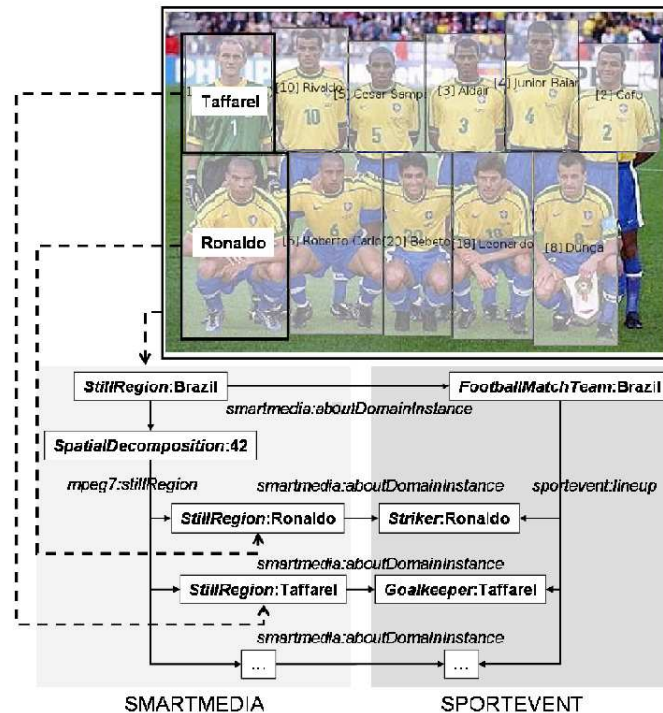


Figure 9.11: A SmartMedia instance representing the decomposition of a picture of the 1998 Brazil World Cup football team.

In order to be able to resolve these types of references, FADE needs to obtain a spatial representation of the content depicted by an image which is displayed on the screen. As outlined in the previous paragraph of this section, images are annotated in SMARTWEB by means of MPEG7-based *StillRegions* which specify bounding boxes and provide an ontological representation of their content. However, these *StillRegions* do not provide spatial information except from their absolute position within the image which means that FADE has to deduce the spatial organization of the objects. In the aforementioned example, the players are grouped into two lines which in turn have spatial characteristics (i. e., the upper vs. the lower line).

The resulting physical environment of type `DoubleColumnList` (see section 6.2.3), consists of two `AbsolutePositions` representing the two lines. The content of these `AbsolutePositions` are physical environments of type `List` with an *eastern* orientation. Thus, in example 131 the built-in reference resolution mechanism of FADE needs first of all to determine the referenced row of the lineup (in this example the lower row). Then it needs to determine the correct starting point (in this example the right side of the row) before the referenced player can be selected by iterating the list of players.

Resolving Multiple Hypotheses

As outlined in section 8.6.6, FADE supports the processing and disambiguation of multiple hypotheses. Multiple hypotheses occur in SMARTWEB if the natural language understanding component is not able to unambiguously interpret a recognized user contribution. In such a case, SPIN produces a set of hypotheses reflecting the different possible interpretations.

The task of FADE is then to select the hypothesis that fits best given the current configuration of the context. This is realized by processing all hypotheses the normal way and by assigning them a score using the scoring mechanism as introduced in section 8.6.6. The PATE plug-in called *resolveMultipleHypotheses* realizes the disambiguation of multiple hypotheses (see algorithm 3). First, it loops over all hypotheses and resolves them via the API of the discourse context. In line 4 (see algorithm 3) it is tested whether the hypothesis represents an elliptical expression and if this is the case the ellipsis is resolved before the resolution of referring expressions is started.

The actual reference resolution takes as argument a container called *EvaluationContainer* that comprises the hypothesis as well as the parameters for scoring it. During the application of *resolveHypothesis*, the referring expressions embedded in the hypothesis are resolved. Furthermore, the collected parameters and the final score are stored in the *EvaluationContainer* object so that this information can be used to sort the resolved hypotheses (see line 10).

Algorithm 3 Resolving Multiple Hypotheses

```

1: TFS[] hypotheses = argument1;
2: Vector resolvedHypotheses = new Vector();
3: for TFS hypothesis : hypotheses do
4:   if isEllipticalExpression(hypothesis) then
5:     hypothesis = resolveEllipsis(hypothesis);
6:   end if
7:   EvaluationContainer evaluation = new EvaluationContainer(hypothesis);
8:   resolvedHypotheses.addAll(resolveHypothesis(evaluation));
9: end for
10: Sort(resolvedHypotheses);
11: return resolvedHypotheses.firstElement();

```

OnFocus/OffFocus Detection

An important task for mobile, speech-driven interfaces that support an open-microphone⁵ is the continuous monitoring of all input modalities in order to detect when the user is addressing the system.

In the mobile scenario of SMARTWEB, the built-in camera of the MDA Pro handheld can be used to track whether a user is present. This camera constantly captures pictures of the space immediately in front of the system. These pictures are processed by a server-side component that detects whether the user is looking at the device or not.

In SMARTWEB, there are two components that determine the attentional state of the user: (i) the OnView/OffView-Recognizer and the (ii) the OnTalk/OffTalk-Recognizer. The task

⁵Open-microphone means the microphone is always active so that the user can interact with the system without further activation. In contrast to an open-microphone interface, systems often require the user to push some hard- or software button in order to activate the system (i. e., a *push-to-activate* button).

of the OnView/OffView recognizer is to determine whether the user is looking at the system or not. The OnView/OffView-Recognizer analyzes a video signal captured by a video camera linked to the mobile device and determines for each frame whether the user is in OnView or OffView mode (figure 9.12 shows two still images of these different modes).

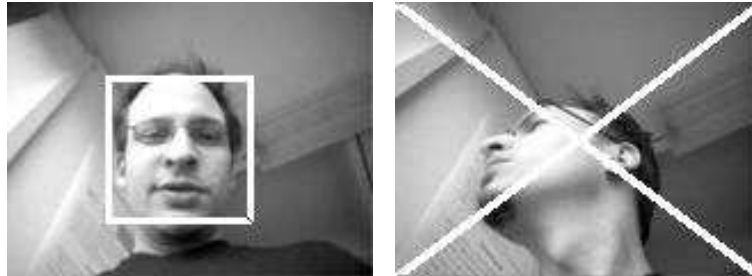


Figure 9.12: Two still images illustrating the function of the OnView/OffView recognizer: The image on the left shows the OnView case and the one the right shows the OffView case.

The task of the OnTalk/OffTalk recognizer is to determine whether a user's utterance is directed at the system. To this end, the OnTalk/OffTalk recognizer analyzes the speech signal and computes about 99 prosodic features based on F0, energy, duration, jitter and shimmer (see Hacker et al. (2006)). This is done for each word but the final result is averaged over the complete turn. Both recognizers provide a score reflecting the individual confidence of a classification. FADE receives and stores the results of the OnView/OffView recognizers as a continuous stream of messages (i. e., every time the OnView state changes, FADE receives an update). OnTalk/OffTalk classifications are only sent to FADE if the speech recognition components detected some input event. Internally, FADE employs an ontological class called *AttentionalStatus* to model the attentional state of the user. This class stores the latest event sent by the OnView/OnTalk recognizers and fuses these to a single state called *onFocus* representing the user's attention (see figure 9.13 for an example).

```
<AttentionalStatus>
<onFocus> false </onFocus>
<onTalk>
<swemma#RecognitionStatus id="pate_17">
  <emma#id> MMR-100261 </emma#id>
  <swemma#onTalk> false </swemma#onTalk>
  <emma#confidence> 1.00 </emma#confidence>
</swemma#RecognitionStatus>
</onTalk>
<onView>
  <swemma#RecognitionStatus id="pate_17">
    <emma#id> MMR-100261 </emma#id>
    <swemma#onView> true </swemma#onTalk>
    <emma#confidence> 1.00 </emma#confidence>
  </swemma#RecognitionStatus>
</onView>
</AttentionalStatus>
```

Figure 9.13: Example of the internal representation of the attentional state for SMARTWEB.

Algorithm 4 shows the basic algorithm for fusing the two classifications into a single state

Algorithm 4 Determining OnFocus/OffFocus

```

1: double onView = getViewRating(); // 1.0 means onView = true; 0.0 = offView
2: double onTalk = getTalkRating(); // 1.0 means onTalk = true; 0.0 = offTalk
3: if onView > 0.3 then
4:   if onTalk < 0.2 then
5:     return offFocus;
6:   else
7:     return onFocus;
8:   end if
9: else
10:  if onTalk < 0.5 then
11:    return offFocus;
12:  else
13:    return onFocus;
14:  end if
15: end if

```

that can be used for further processing. The overall idea is to combine the two distinct classifications for OnView/OffView and OnTalk/OffTalk in order to compensate for potential classification errors. If the OnView value is above 0.3 (where 0 means OffView and 1 means OnView), the OffTalk value must be very low (below 0.2) in order to classify a contribution as OffFocus (see lines 3-5). Otherwise, a OnTalk value below 0.5 is already sufficient to classify an utterance as OffFocus.

Since this functionality is not part of the standard FADE API, it is realized by means of a PATE plug-in called *CheckAttentionalStatus*. This plug-in is used by all PATE rules that deal with incoming OnView or OnTalk data. It analyses the latest OnView and OnTalk events and updates the **onFocus** slot of the **AttentionalStatus** according to the algorithm depicted in algorithm 4. This **onFocus** slot in turn is used by all rules that pass the final results of FADE on to the subsequent modules. If **onFocus** is set to false, these rules can not fire and appropriate fallback mechanisms take over.

9.4 FADE in the OMDIP System

In this section we describe FADE's basic functionality within the OMDIP system and discuss the processing strategy of FADE. OMDIP is a multimodal dialogue system and since it is a rather prototypical but less complex dialogue system, we will also describe in detail what happens during a complete interaction loop, i. e., from the beginning of a user utterance until the system finished its response.

9.4.1 The OMDIP System

The demonstration scenario of OMDIP is a multimodal web-based platform for selling different items like *Soundlogos* or musical greetings. Soundlogos replace the normal ring-tone that callers hear until the callee picks up. Nowadays, it is possible to play complete recordings of popular songs. The purpose of this application is that users of a mobile phone can easily change their current Soundlogo themselves. This involves several sub-actions such as: listen-



Figure 9.14: The OMDIP system in action: The user selects a *Soundlogo* using the pen while uttering the request to play the selected Soundlogo.

ing to the Soundlogo, asking for pricing information, and activating a particular Soundlogo. A musical greeting is a recorded verbal message accompanied by music that can be sent via the classic telephone system to a receiver at a specified time.

In 2006 we developed two versions of the OMDIP system; the first deals with the presentation and selling of Soundlogos, and the second with the presentation and selling of musical greetings. In what follows, we will only discuss the first system since it shows a reduced complexity that permits a more detailed discussion of the actual processing in FADE.

The *SoundLogo* Scenario

The demo-application of OMDIP is a multimodal user interface for mobile devices to a so-called campaign manager for a telecommunications company (see figure 9.14). The application offers a number of services which can be accessed by stylus or by voice. The campaign manager can only be accessed by registered customers. This means that users need to first log in with their mobile phone number and PIN.

Since the following use-case centers around the Soundlogo services all other options are not discussed any further here. A Soundlogo is a piece of music combined with an ordinary ring back tone. That means a phone caller who tries to call a number with activated Soundlogo hears the configured music instead of the normal ring back tone. Figure 9.14 shows the overview page of the available Soundlogos.

The following interaction sequence reflects a typical interaction with the Soundlogo application of the OMDIP system:

- (133) **System:** [shows overview of available soundlogos]
- (134) **User:** *“Play this one.”* [pointing gesture at the Soundlogo ‘Perfekte Welle’ with the pen]
- (135) **System:** [plays the soundlogo ‘Perfekte Welle’]
- (136) **User:** *“What is the price of that soundlogo?”*
- (137) **System:** *“The price of ‘Perfekte Welle’ is 1.99 Euro.”*
- (138) **User:** *“Okay, activate it.”*
- (139) **System:** *“The Soundlogo ‘Perfekte Welle’ has been activated. What else do you want to do?”*
- (140) **User:** [pointing gesture to the link “Service-Zentrum”]
- (141) **System:** [displays the page “Service-Zentrum”]

9.4.2 The Architecture of the OMDIP System

The target platform of the OMDIP system is an MDA III PDA that is connected to the dialogue system via a HTTP connection. The dialogue system is embedded in a Web environment (Tomcat) and runs on a remote server.

The dialogue manager of the OMDIP system pretty much resembles the architecture of SMARTWEB with the exception that the REAPR component has been exchanged by a full-fledged dialogue manager. In total, the system consists of seven components that are integrated into the IHub infrastructure of SMARTWEB (see section 9.3).

Automatic Speech Recognition (ASR): The ASR of the first version of OMDIP is located on the server while the ASR of the second version of OMDIP is located on the mobile device.

Speech Interpretation: The speech interpretation component is responsible for the parsing recognized speech input. Just as in SMARTWEB, this component is realized by using the SPIN component (see Engel (2005, 2006)).

Gesture Recognition: The gesture recognizer is located at the mobile device. It supports the recognition of pointing gestures which are resolved to a unique identifier that is sent to FADE.

Fusion and Discourse: The fusion and discourse engine is realized by means of FADE.

Action Planner: The action planner is responsible for managing the dialogue and accessing the function modeler. This component is a successor of the action planner of the VIRTUALHUMAN system (see Löckelt (2007)).

Function Modeler: The function modeler encapsulates the backend application of OMDIP. In addition to that, it also provides access to a database where the individual songs are stored.

GUI Generation: The GUI generation component is responsible for the graphical layout of the system response. In the second version of OMDIP this component is realized by means of the NIPSGEN component of SMARTWEB.

Speech Generation: The speech generation component realizes the actual speech output of the system. It takes abstract ontological instances and selects appropriate system prompts.

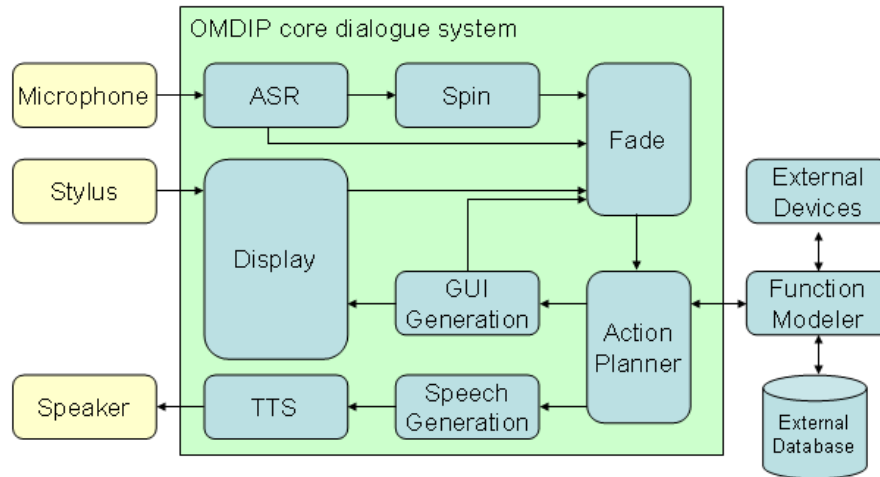


Figure 9.15: The basic architecture of the OMDIP system.

Tasks for FADE

FADE fulfills the following tasks in the OMDIP system:

Multimodal integration

(142) **User:** “*Activate this [pointing gesture] Soundlogo.*”

Here, FADE has to fuse the output of the speech recognizer and the gesture recognizer in order to obtain an integrated multimodal interpretation of the user’s utterance. This functionality is realized by means of the built-in methods for reference resolution.

Reference resolution

(143) **User:** “*Activate the third Soundlogo.*”

(144) **User:** “*Activate this Soundlogo.*”

The resolution of referring expressions is also dealt with by the built-in API of the discourse modeler.

Resolution of elliptical expressions

(145) **User:** “*What does the first Soundlogo cost?*”

System: ...

User: “*And the second one?*”

- (146) **User:** “What does the first Soundlogo cost?”
System: ...
User: “And this one?” [pointing gesture]

Here, FADE needs first to resolve the referring expressions that are embedded in the elliptical construction (e. e., “the second one”) before the resolution process for the elliptical construction can take place. Again, this is realized by means of the built-in functionality of FADE’s API.

9.4.3 FADE’s Interfaces in OMDIP

FADE receives input from four components of OMDIP:

1 SR_STATUS (ASR):

- a. Every time the speech recognizer detects an input signal, it sends a *StartOfRecognition* message to FADE:

```
<object type="emma#Info">
  <slot name="swemma#status">
    <object type="swemma#ASRStatus">
      <slot name="swemma#dispatch">
        <object type="swemma#StartOfRecognition"/>
      </slot>
    </object>
  </slot>
</object>
```

- b. If the ASR fails to process the input signal, it sends the following:

```
<object type="emma#Info">
  <slot name="swemma#status">
    <object type="swemma#ASRStatus">
      <slot name="swemma#dispatch">
        <object type="swemma#RecognitionError"/>
      </slot>
    </object>
  </slot>
</object>
```

- c. If the user did not say anything, the ASR sends the following:

```
<object type="emma#Interpretation">
  <slot name="emma#noInput">
    <value type="string"> true</value>
  </slot>
</object>
```

- d. If the user said something that could not be analyzed, the ASR sends the following:

```
<object type="emma#Interpretation">
  <slot name="emma#uninterpreted">
    <value type="string"> true</value>
  </slot>
</object>
```

- 2 SPIN:** SPIN sends the analysis results (called interpretation hypotheses) for the ASR output to FADE. The content of the different interpretation hypotheses for the individual use cases is described in section 9.4.1.

- 3 GR_INPUT (Gesture recognition):** The gesture recognition component sends the ID of the object the user clicked on to FADE. The content of the message looks like this:

```

    <emma:interpretation id="GR-1-1"
      emma:start="1112271714873"
      emma:end="1112271714873"
      emma:confidence="1.0"
      emma:mode="ink">
      <discourse:PointingGesture>
        <discourse:objectReferenceString>
          http://smartweb.semanticweb.org/individuals/ap#soundlogo02
        </discourse:objectReferenceString>
      </discourse:PointingGesture>
    </emma:interpretation>

```

- 4 **GUL_GEN**: Every time the displayed content changes, **GUL_GEN** sends the updated display representation to **FADE**. This message comprises both a flat representation of the clickable objects and a more complex representation of the graphical organization of the objects (e.g., list representations). An example of such a message looks like this:

```

<discourse:Display>
  <discourse:has_clickableObject>
    <discourse:Service rdf:resource="http..."/>
  </discourse:has_clickableObject>
  <discourse:has_physicalEnvironment>
    <discourse:PhysicalEnvironment>
      <discourse:has_PEType>
        <discourse:List/>
      </discourse:has_PEType>
      <discourse:has_orientation> southern </discourse:has_orientation>
      <discourse:has_absolutePosition>
        <discourse:AbsolutePosition rdf:about="http://.../individuals/gui#01">
          <discourse:has_contentObject>
            <tsys:Soundlogo rdf:resource=".../individuals/ap#soundlogo01"/>
          </discourse:has_contentObject>
          <discourse:has_southernNeighbor>
            <discourse:AbsolutePosition rdf:resource=".../individuals/gui#02"/>
          </discourse:has_southernNeighbor>
        </discourse:AbsolutePosition>
      </discourse:has_absolutePosition>
      <discourse:has_absolutePosition>
        <discourse:AbsolutePosition rdf:about=".../individuals/gui#02">
          <discourse:has_contentObject>
            <tsys:Soundlogo rdf:resource=".../individuals/ap#soundlogo02"/>
          </discourse:has_contentObject>
          <discourse:has_northernNeighbor>
            <discourse:AbsolutePosition rdf:resource=".../individuals/gui#01"/>
          </discourse:has_northernNeighbor>
          <discourse:has_southernNeighbor>
            <discourse:AbsolutePosition rdf:resource=".../individuals/gui#03"/>
          </discourse:has_southernNeighbor>
        </discourse:AbsolutePosition>
      </discourse:has_absolutePosition>
      <discourse:has_absolutePosition>
        <discourse:AbsolutePosition rdf:about=".../individuals/gui#03">
          <discourse:has_contentObject>
            <discourse:Soundlogo rdf:resource=".../individuals/ap#soundlogo03"/>
          </discourse:has_contentObject>
          <discourse:has_northernNeighbor>
            <discourse:AbsolutePosition rdf:resource=".../individuals/gui#02"/>
          </discourse:has_northernNeighbor>
        </discourse:AbsolutePosition>
      </discourse:has_absolutePosition>
    </discourse:PhysicalEnvironment>
  </discourse:has_physicalEnvironment>
</discourse:Display>

```


9.4.4 The Processing Strategy of FADE in OMDIP

As discussed in section 8.5, the general idea is to partition the complete processing of FADE into three phases where each phase is represented by an object on the goal-stack: (i) data-collection—represented by the working memory element (WME) *SynchronizationPhase* on top of the goal-stack, (ii) interpretation—represented by the WME *InterpretationPhase*, and (iii) evaluation—represented by the WME *EvaluationPhase*. Every time FADE receives some input event, this event is pushed onto PATE’s goal-stack from which it can be accessed by the production rules. First, FADE is in the data-collection phase where it collects incoming input events from the individual modalities and tries to identify turn boundaries. This phase is realized through a set of specialized production rules for PATE that performs the required actions, e. g., *pushTimeOut* (plus variations) and all rules that start with *handle...* (e. g., *handlePointingGesture*).

In order to be able to identify the end of a turn as soon as possible, FADE needs to know whether there are unprocessed input events in the recognition pipelines at any point in time. Therefore, FADE expects the recognition components to send a *startOfRecognition* message that should be sent at the very moment the recognition component detects the input event. When a user turn starts, FADE waits for the first completed *startOfRecognition/recognition-result* pair and then starts a time-out which lasts 500 msec.; specified in the respective rule, e. g., *pushTimeOut(2)*⁶. During the time-out, FADE waits for additional input events but considers the turn to be completed if the time-out has run out without new input events to occur⁷. The synchronization phase ends if there is no *startOfRecognition* event left and at least some input event (or a *NotUnderstood* dialogue act) has been recognized. In the next phase, PATE interprets the perceived events and tries to resolve any contained elliptical or referring expressions (this is done by the rule *handleReferences*). In the final phase, either the rule *outputResult* sends the integrated interpretation hypothesis to the action planner, or one of the *outputNotUnderstood* rules passes on the *NotUnderstood* dialogue act to the action planner. Eventually, the rules starting with *CleanUp* will remove any remaining pointing gestures or interpretation hypotheses from the WM.

Processing Incoming Monomodal Events

Every incoming monomodal input event is pushed to the goal-stack of FADE for further processing. However, if the goal-stack is empty, two additional WMEs will be pushed to the goal-stack: (i) the WME *InterpretationPhase* and (ii) the WME *EvaluationPhase*. These two WMEs control the aforementioned processing strategy of FADE.

The Rule Base

For the first version of the OMDIP system, we developed 31 PATE rules that deal with the individual monomodal input events generated by SPIN, the gesture recognizer and the GULGEN component. Note that not all rules are actually used in a normal system run. Only

⁶This time-out is necessary since the sometimes difficult handling of the pen and the mobile device causes users to enter their commands in a sequential way (see also section 4.1.2). In the VIRTUALHUMAN system, for example, FADE does not employ any timeout after a recognition result has been received.

⁷Note that the time-out of 500 msec. is tailored to the delay caused by the speech recognizer so that the complete processing delay will never exceed two seconds (which marks the boundary for acceptable reaction times of human-computer interfaces)

a smaller subset of the rules is used under normal conditions (in appendix B a small excerpt of the key rules FADE's rule base for the OMDIP system is depicted). The purpose of the remaining rules is to cover potential errors, e. g., if FADE receives a speech analysis result from SPIN but did not receive any *startOfSpeech* message beforehand. In such a case, the rule *handleInterpretation* would not be able to process this interpretation. Then, however, the rule *handleInterpretation(WithoutStartOfSpeech)* fires and FADE remains in a consistent state.

9.4.5 Step-by-Step Example of the Application of Rules

In the following subsections we will give three step-by-step examples of how FADE processes several input events. We will start with the initial configuration and the processing of the display representation. Then we will show how a *StartOfRecognition* event is processed and finally we will show how a more complex multimodal command is processed.

Initial Configuration of the WM before a turn starts

Before a turn starts, the WM of FADE comprises two WMEs: i) the representation of the physical environment (WME3 in figure 9.16) and ii) the display representation (WME2 in figure 9.16). This initial state is triggered through the GUI_GEN component which sends the current display representation to FADE. This object is pushed to the goal-stack along with two other WMEs that are pushed to the goal-stack before any incoming input event. These three objects cause the following rules to fire:

```
PATE-Thread 0 > Rule initializeDisplayRepresentation(2) [0.5] fires! with poS 0.28125
PATE-Thread 0 > Rule popInterpretationPhase [0.3] fires! with poS 0.16249999999999998
PATE-Thread 0 > Rule popEvaluationPhase [0.1] fires! with poS 0.05416666666666667
```

Processing a StartOfRecognition Event

If the speech recognizer detects some input signal, it immediately informs FADE by sending a *StartOfRecognition* event. Again, this event is pushed to the goal-stack of FADE along with a WME *InterpretationPhase* and a WME *EvaluationPhase* (see figure 9.16). This causes the following rules to fire (taken from the system trace):

```
PATE-Thread 0 > Rule handleStartOfSpeech [1] fires! with poS 0.5541666666666667
PATE-Thread 0 > Rule popInterpretationPhase [0.3] fires! with poS 0.16249999999999998
PATE-Thread 0 > Rule popEvaluationPhase [0.1] fires! with poS 0.05416666666666667
```

Afterwards, the configuration of the WM comprises three objects in the WM and no object on the goal-stack.

Processing a Multimodal Command

A multimodal command consists of a sequence of input events: First, the user starts to speak and FADE receives a *StartOfRecognition* event from the speech recognizer (see the previous subsection for a description of how this event is processed). Then, FADE receives two interpretations: One for the spoken command and one for the pen input in an arbitrary sequence. Here, we will assume that FADE first receives the pen input which is processed in two steps:

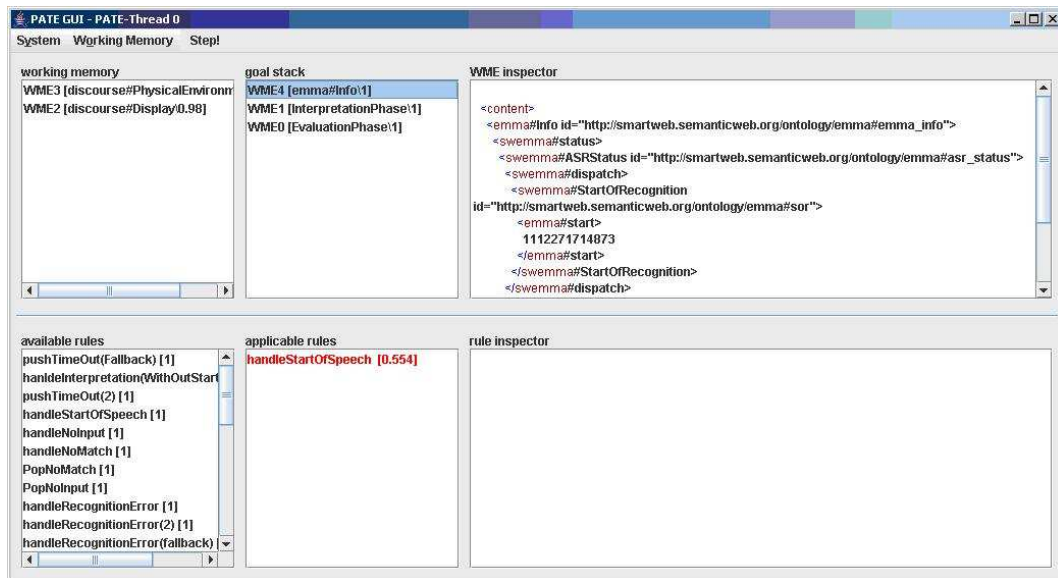


Figure 9.16: OMDIP: A StartOfRecognition event on top of the goal-stack.

First, the String reference to the selected object is resolved by the rule *resolvePointingGesture* (see figure 9.17), then, the rule *handlePointingGesture* updates DIM (i.e., the pointing gesture is added to the context representation) and pops the pointing gesture from the goal-stack. Due to the fact that there is now a pointing gesture in the WM, the two WMEs *InterpretationPhase* and *EvaluationPhase* remain on the goal-stack and will not be removed. The next input event that is sent to FADE is now the result of SPIN. This input event is handled by the rule *handleReferences* fires. This rule triggers the contextual interpretation of received speech interpretation via the plug-in *ResolveReferringExpressions*. The API of the discourse context then resolves (if possible) all embedded referring expressions, tries to incorporate the received pointing gestures and returns the resolved interpretation. The rule *handleReferences* then replaces the original interpretation with the resolved one and removes the WME *InterpretationPhase* from the goal-stack.

Now there is only the WME *EvaluationPhase* left on the goal-stack. Normally, this phase triggers the evaluation of the different interpretation hypotheses against each other. However, as the first version of OMDIP only considers a single hypothesis, the rule *outputResult* directly triggers the output to the action planner. Besides this, the rule *outputResult* also updates the context representation with the resolved hypothesis. Eventually, the WME *EvaluationPhase* is popped from the goal-stack and a WME *CleanUp* is pushed onto it. This WME triggers the *Cleanup* rules which remove all remaining interpretation fragments from the WM. The complete trace of firing FADE rules for this interaction looks like this:

```
PATE-Thread 0 > Rule handleStartOfSpeech [1] fires! with poS 0.5541666666666667
PATE-Thread 0 > Rule popInterpretationPhase [0.3] fires! with poS 0.16249999999999999
PATE-Thread 0 > Rule popEvaluationPhase [0.1] fires! with poS 0.054166666666666666
PATE-Thread 0 > Rule resolvePointingGesture [1] fires! with poS 0.37166666666666666
PATE-Thread 0 > Rule handlePointingGesture [0.5] fires! with poS 0.27354166666666666
PATE-Thread 0 > Rule handleReferences [0.5] fires! with poS 0.30416666666666667
PATE-Thread 0 > Rule outputResult [0.2] fires! with poS 0.10872222222222222
```

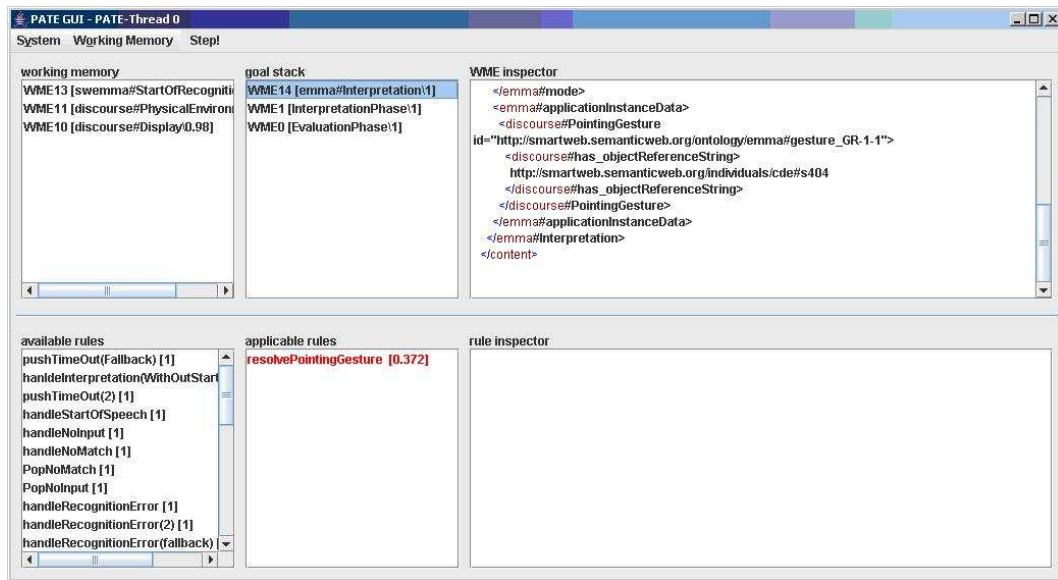


Figure 9.17: OMDIP: Configuration of the working memory and the goal-stack just before a pointing gesture is processed.

```
PATE-Thread 0 > Rule cleanUp [1] fires! with poS 0.5379166666666666
```

9.4.6 Changes in the Second Version of OMDIP

As mentioned at the beginning of this section, the second version of the OMDIP system differs from the first version in some aspects. From the perspective of FADE, the processing of user contributions has changed in two important aspects: (i) instead of a single interpretation hypothesis, the recognition and analysis components (i. e., ASR, SPIN and FADE) now exchange multiple interpretation hypotheses and (ii) the action planner now provides expectations about the likely and possible user actions given the current state of the conversation.

To this end, the processing strategies of FADE had to be revised. However, the new processing strategies were completely realized by updating the PATE rule base of FADE which underlines the overall toolkit character of FADE. In order to be able to deal with multiple hypotheses, we only had to replace the original plug-in “ResolveReferences” for accessing the discourse modeler API with the plug-in that is used in SMARTWEB (called “ResolveMultipleHypotheses”). The expectation-based processing required some modifications in the synchronization of input and output events which required 8 new rules.

9.5 Summary

In this chapter we have discussed three multimodal dialogue systems that serve as a proof of concept for our integrated multimodal fusion and discourse engine. We have outlined how FADE has been applied in these quite diverse systems in order to fulfill a variety of tasks. We have also shown that the effort that has to be made in order to integrate FADE into such a system is comparatively low. Because of the flexible rule engine of PATE, adaptations or

adjustments can be realized without changes to the code base of FADE. Table 9.2 provides an overview of the individual discourse phenomena that are supported by FADE for the three systems.

Discourse phenomenon	VirtualHuman	SmartWeb	OMDIP
anaphoric references	X	X	X
discourse deixis	X		X
place deixis	X	X	X
spatial references	X	X	
temporal references		X	X
indirect references		X	
exophoric references		X	X
elliptical expressions	X	X	X
turn-taking management	X		
addressee identification	X		
reactive feedback	X		

Table 9.2: Overview of the individual discourse phenomena as they are covered by FADE in the VIRTUALHUMAN, SMARTWEB and OMDIP systems.

An important aspect here is, that most of the functionality can be realized by using the built-in methods of the discourse context API. Furthermore, we have shown that it is also possible to realize quite complex functionality that can be seamlessly incorporated into FADE's functionality by using PATE rules. An example of this is the realization of the OnFocus and OffFocus detection in SMARTWEB (see section 9.3.3). Here, the complete processing logic for dealing with the asynchronous OnTalk/OffTalk and OnView/OffView messages as well as the calculation of the resulting OnFocus/OffFocus status has been realized by means of PATE rules and a single plug-in. With the OMDIP system, we have also shown that FADE can be employed in a commercial setting, i. e., it supports a robust and efficient processing. The VIRTUALHUMAN system serves as a proof of concept that FADE is not only capable of dealing with dyadic discourse, but that it also supports the processing of interactional contributions as they arise in multiparty interactions.

	SmartWeb	VirtualHuman	OMDIP
Number of production rules	18	30	31
Number of plug-ins	5	9	2
Ratio built-in generic API functions vs. application specific rules	80:20	60:40	70:30

Table 9.3: Overview of FADE's rule base for the SmartWeb, VirtualHuman and OMDIP systems.

An important aspect of FADE is its generic orientation which permits a straightforward integration of FADE into new dialogue systems or applications. Depending on the tasks at hand, only little adaptation and implementation work has to be done in order to integrate FADE into a new system. Table 9.3, for example, shows the amount of production rules and plug-ins that have been implemented in order to adapt FADE to the SMARTWEB VIRTUALHUMAN and OMDIP system. However, the more additional tasks FADE has to fulfill that are

not covered by the standard API of, the more functionality has to be realized by specialized production rules. In SMARTWEB for example, about 80 % of the functionality is realized by means of FADE's standard API and the standard production rules for the three processing phases as discussed in section 8.6. Only the additional task of fusing the OnTalk and OnView messages requires a small set of six additional rules and one plug-in. VIRTUALHUMAN in contrast, requires more system-specific rules in order to be able to realize the turn-taking and reactive behavior of the virtual characters so that only 60% of the functionality are realized here by means of the built-in functionality.

User Input	Number of Interpretation Hypotheses	Processing Time (in msec.)
Simple monomodal input		
User: <i>"Show me the weather forecast for tomorrow."</i>	2	61
User: <i>"Who won the World Cup in 2006?"</i>	1	38
Context-dependent monomodal input		
User: <i>"How often did this team win?"</i>	1	46
User: <i>"And Germany?"</i>	1	32
User: <i>"What is the name of the third player from the upper left?"</i>	1	35
User: <i>"How can I get there."</i>	1	29
Context-dependent multimodal input		
User: <i>"Give me more information about this."</i> [+ pointing gesture at a restaurant displayed on the screen]	4	79
User: <i>"How often did this player score?"</i> [+ pointing gesture at a football player displayed on the screen]	2	31

Table 9.4: Overview of FADE's processing times for a selection of sample input taken from the SMARTWEB system.

Figure 9.4 provides an overview of the performance of FADE for sample user input taken from the SMARTWEB system. In general, the performance of FADE depends on the complexity of the exchanged data, the type of the interaction and the current load of the computer. Unimodal pointing gestures, for example, require nearly no additional processing time and achieve the fastest reaction times. Multimodal input (e.g., a deictic reference accompanied by pointing gestures) require some additional processing but are processed faster than context-dependent multimodal input. There is also a dependency between the number of interpretation hypotheses generated by the natural language understanding component and the processing time of FADE. Overall, FADE shows fast reaction times that allow the application of FADE in multimodal dialogue systems that aim at real-time interactions.

Chapter 10

Conclusion and Discussion

This chapter aims at summarizing the work done in this thesis. It highlights again the major scientific contributions and underlines the importance of the findings made in this thesis. The chapter concludes this thesis by providing directions for future research.

10.1 Contributions

Previous research in the area of multimodal dialogue systems has focused mainly on specialized approaches to multimodal fusion and discourse processing that are only capable of dealing with specific subsets of multimodal contributions. Recent approaches to multimodal fusion, for example, have mostly concentrated on the integration of gestures and speech input on the propositional level. This means that interactional contributions (e. g., turn-taking signals) cannot be processed by these approaches. With respect to the modeling of the context, current approaches still focus either on the representation of the discourse context or the physical context. This is done, however, neither in an integrated fashion, nor do these approaches incorporate the immediate conversational context, i. e., the participants' current roles and displayed nonverbal behavior.

In this thesis we have developed a generic and comprehensive context model that supports the integration of perceived monomodal events into a multimodal representation, the resolution of referring expressions and the generation of reactive and deliberative actions. Besides the classic linguistic context provided by a discourse history, we have identified sets of physical and conversational context factors that a multimodal dialogue system aiming at real conversational interaction needs to incorporate. Based on this extended notion of context, we have presented an integrated approach to multimodal fusion and discourse processing called FADE. We have shown how FADE supports the processing of both natural nonverbal and verbal behavior in dyadic as well as multiparty conversations.

The integration of FADE into three distinct dialogue systems has proven the general applicability of this approach for a wide range of multimodal dialogue systems. Within these systems, FADE is responsible for the resolution of referring expressions (e. g., anaphoric references, exophoric (cross-modal) references, place deictic references, spatial references and references), the mutual disambiguation of interpretation hypotheses, the resolution of elliptical or fragmental expressions, the identification of the intended addressee(s) and the processing and triggering of reactive behavior (e. g., turn-taking signals). Through the application of FADE in these diverse systems we have been able to show that FADE can be used in an

efficient and flexible way. In the running systems, FADE's robustness and promptness of reaction has been demonstrated.

10.1.1 Main Scientific and Practical Results

The main scientific and practical results can be divided into three areas: (i) theoretical analysis and modeling, (ii) algorithms and architecture and (iii) practical aspects.

Theoretical Analysis and Modeling

- **A detailed analysis of context factors for multimodal multiparty dialogue systems:**

Previous work in the area of multimodal dialogue systems has focused only on a limited set of context-dependent discourse phenomena. In Chapter 2 and 3, this thesis has provided a detailed analysis of verbal and nonverbal context-dependent phenomena that need to be covered by the current and next generation of conversational dialogue systems. Based on this analysis, we have defined the requirements for the comprehensive context model of FADE.

- **A comprehensive context model for multimodal multiparty dialogue systems:**

In Chapter 6, we have introduced a comprehensive model of the context of dyadic and multiparty conversations that extends the standard notion of a discourse context with a representation of the immediate conversational context. This conversational context encompasses detailed representations of the physical surroundings, of the participants including their displayed nonverbal behavior, and the current and past conversational roles of the participants. This context model provides the basis for the development of FADE.

Architecture and Algorithms

- **An architecture for dealing with the full range of multimodal discourse phenomena:**

Based on the comprehensive context model developed in Chapter 6, we have introduced a new architecture for the context-based multimodal interpretation in multimodal dialogue systems. The key to this architecture is that it differentiates between interactional and propositional content of contributions. This distinction allows for the reactive interpretation of the interactional aspects of contributions (e. g., turn-taking signals) and the interpretation of context-dependent propositional content like referring or elliptical expressions.

- **A generic and reusable component for multimodal fusion and discourse processing:**

With FADE we have introduced a generic toolkit for dealing with context-dependent phenomena in multimodal multiparty dialogue systems. Central to FADE is that it supports the processing of a variety of context-dependent multimodal discourse phenomena such as the resolution of referring and elliptical expressions, the identification of the intended addressee and the processing and processing of turn-taking signals or reactive behavior. The integration of FADE into three diverse dialogue systems has

shown that it meets the requirements specified in the introduction namely efficiency, flexibility, robustness and promptness of reaction.

- **An integrated approach to multimodal fusion and discourse processing:**

A key aspect of the reference resolution algorithm that is presented in this thesis is the integration of two usually separated tasks, i.e., multimodal fusion and reference resolution. Typically, these two tasks are performed in a sequential fashion. In this thesis, however, we have introduced an extended reference resolution algorithm that is capable of performing both tasks at the same time. Moreover, this algorithm is capable of processing ambiguous deictic input that can either be resolved by means of a pointing gesture or a referent that has been introduced during the previous discourse.

Practical Aspects

- **FADE—a generic toolkit for multimodal fusion and discourse processing**

The FADE framework described in this thesis realizes a generic and efficient toolkit that has been employed in a variety of multimodal dialogue systems. It supports the resolution of a variety of referring expressions like anaphoric references, discourse deixis, place deixis, spatial references, temporal references, indirect references, exophoric references and elliptical or fragmental expressions. Moreover, it supports the interpretation and generation of turn-taking and reactive behavior for multiparty interactions. A highlight of FADE is that most of its functionality can be used without modification and that the system-specific functionality can be realized by means of production rules and plug-ins without changing the code-base. Thus, FADE can be easily integrated into new applications or dialogue system by application developers. Moreover, the efficient implementation of FADE accounts for fast processing times even for complex context-dependent input. Thus, FADE realizes a toolkit that contributes to the initial claim for bridging the performance gap between commercial and research dialogue systems.

- **An efficient and flexible approach for the representation and manipulation of ontological data:**

With the eTFS framework we have introduced an efficient and flexible framework for the representation and manipulation of ontological data. In contrast to existing approaches like the Jena¹ framework, the eTFS framework provides a fast and light-weighted representation of complex data that is compatible to common ontological representation formats like RDF/S, OWL, or OIL. Moreover, eTFS encompasses efficient implementations of operations like unification and overlay.

- **The production rule system PATE**

With the production rule system PATE, we have introduced a generic component that supports the rapid prototyping of a wide range of components for multimodal dialogue systems. The PATE system has become a robust toolkit for realizing a variety of different tasks. Thus, it has not only been successfully employed in FADE but also in

¹The Jena toolkit is an open source framework for building Semantic Web applications and it is available for download at <http://jena.sourceforge.net>. Jena provides a rich API for accessing, storing and manipulating RDF, RDFS and OWL based knowledge representations. It is currently one of the most widely distributed tools for dealing with RDF or OWL based data. However, because of the rich reasoning capabilities and the complex internal data-model, Jena is not efficient and can cause serious delays which disqualifies the framework for use in real-time applications.

other systems. Examples for this application of PATE in other systems are the TALK project where it has been employed to realize the core components of the SAMMIE system, the VIRTUALHUMAN project where it has been used to realize the natural language understanding component and the multimodal generator and SMARTWEB where it has been used to realize the *Motorbike Danger Warning Environment* component.

In addition to this, we have been able to successfully employ the developed approach in three different multimodal dialogue systems, a limited domain dyadic dialogue system, an open domain dyadic dialogue system and a multiparty dialogue system where up to two humans can interact with three virtual characters. A key characteristic of this approach is that all adjustments and system specific processing strategies can be realized without changes in the code base of FADE. This underpins the toolkit character of FADE.

10.1.2 Publications

This section lists the publications resulting from the research for this thesis.

- **Journals and books**

Parts of this work have been presented in the Journal “Conversational User Interfaces” by Wahlster (2004a) (see Pflieger and Alexandersson (2004)), the “International Journal of Virtual Reality” (Löckelt et al., 2007), in the book “SmartKom—Foundations of Multimodal Dialogue Systems” by Wahlster (2006) (see Alexandersson and Pflieger (2006); Engel and Pflieger (2006); Alexandersson et al. (2006)) and in the book “Artificial Intelligence for Human Computing” (Sonntag et al., 2007b).

- **Conferences**

Parts of this work have been presented at the following international conferences: The *International Conference on Multimodal Interfaces* (ICMI) (Pflieger, 2004, 2005; Reithinger et al., 2006), the *International Conference on Intelligent Virtual Agents* (IVA 2006) (Pflieger and Löckelt, 2006), the *European Conference on Speech Communication and Technology* (Eurospeech) (Pflieger and Löckelt, 2005), the *International Conference on Spoken Language Processing* (Interspeech-ICSLP) (Pflieger and Schehl, 2006), the *Conference on INtelligent TEchnologies for interactive enterTAINment* (INTETAIN) (Löckelt et al., 2005), the *International Conference on Technologies for Interactive Digital Storytelling and Entertainment* (TIDSE) (Löckelt and Pflieger, 2006) and *Konferenz zur Verarbeitung natürlicher Sprache* (Konvens) (Pflieger et al., 2002; Alexandersson et al., 2004).

- **Workshops**

Parts of this work have been presented at the following international workshops: At the workshop series *Semantics and Pragmatics of Dialogue* (Pflieger and Alexandersson, 2006; Löckelt and Pflieger, 2005; Pflieger et al., 2003b; Löckelt et al., 2002), at the workshop on *Knowledge and Reasoning in Practical Dialogue Systems* (Pflieger et al., 2003a) and at the workshop on AI for Human Computing (AI4HC’07) (Sonntag et al., 2007a).

10.2 Future Work

The scope of this thesis has not allowed us to deal with every aspect of the topic in the detail it deserved. In the following we will provide possible directions for future research:

- **Backchannel Feedback**

Backchannel feedback has been introduced as feedback provided by the listeners to inform the speaker of their current understanding of the ongoing turn. Backchannels can be expressed both through verbal behavior with utterances like “*yes*”, “*ok*” or “*hmm*” and through nonverbal behavior like head nods or facial expressions. An interesting observation is that speakers seem to implicitly request backchannel feedback from their audience as they organize their contributions in so-called *installments*. These installments are separated by short pauses that invite the addressees to provide some feedback.

Up to now, FADE has been used in VIRTUALHUMAN to realize gaze behavior and head nods in order to support the turn-taking system. The next step, however, is to add backchannel feedback that contributes to the grounding mechanism. For this, a light-weighted component would be required that is able to identify short pauses in the speech signal of the user. The virtual characters in turn could use this information to generate appropriate backchannel feedback (e. g., slight head nods or facial expressions) depending on their current affective state and their understanding of the conversational state.

- **Adaptation to Multimodal Interaction Patterns**

In section 4.1.2 we discussed the different types of multimodal integration patterns that can be observed when human users interact with multimodal dialogue systems. Users of current multimodal dialogue systems tend to exhibit stable integration patterns. There is common agreement that future multimodal systems should be able to detect and adapt to the way the users provide their input. This is expected to increase system robustness and processing speed as the individual time-outs can be optimized during the interaction. Using our context-based approach it would be easy—if a predominant integration pattern has been recognized—to incorporate this information into the context model and use it for further integration.

- **Detection and Processing of Irony**

Understanding irony or sarcasm in a dialogue system is a complex task since it requires the robust detection of a sarcastic tone. Tepperman et al. (2006) present some experiments toward sarcasm recognition using prosodic, spectral and contextual cues. Their results suggest that sarcasm can be detected by means of a combination of spectral and contextual cues as well as a human annotator would and conclude that prosody alone is not sufficient to determine whether a speaker is sarcastic. In their conclusion they even suggest concentrating on contextual features:

Designers of a dialogue manager or dialogue transcriber wishing to detect sarcasm would probably be best served to just ignore prosodic features entirely and concentrate on extracting the kinds of contextual features we used in this study. Most of them would be simple enough to detect automatically, either from acoustics (for laughter, gender, pause) or semantics and syntax (question/answer, start/end). (Tepperman et al., 2006, p. 1841)

Taking the affective and emotional aspects of a participant into account is expected to improve the processes of interpreting user contributions and error recovery (Streit et al., 2006). Even though the representation of the emotional and affective state is supported by the context model developed for this thesis, it is not possible to detect phenomena like irony or sarcasm using the built-in functionality of the current version of FADE. Future research in that area consists of the development of a generic model and implementation of the identification of ironic or sarcastic expressions.

- **Incremental Processing**

Incremental processing of user contributions is crucial when aiming at a dialogue system that supports full-fledged reactive behavior (i. e., display and interpretation of backchannel feedback). However, it is the task for future research to extend FADE so that it is able to provide incremental output. Of particular interest here is the question of how the incremental output of FADE can be represented, and how the processing of the subsequent components will thereby be affected. Even though the immediate conversational context is already able to deal with updates on a frequent basis, it needs to be investigated as to whether synchronization issues might occur.

- **Integration of FADE into the AMADA system**

We are currently in the process of integrating FADE into a new multimodal dialogue system called AMADA. This system is a successor of the SAMMIE system that was developed in the TALK project (see section 7.7.1). The AMADA system is completely realized by means of PATE components which means that even the middle-ware for controlling the information exchange between individual components is realized by a PATE-based component (in the SAMMIE system the middle-ware was realized by means of the Open Agent Architecture (OAA); see Martin et al. (1999)). The primary focus of this system is to realize a multimodal human-computer interface that shows the same reaction times as mouse-based interfaces and the first prototype already shows reaction times below 500 msec. Thus, the AMADA system is the continuation of the efforts of this thesis to close the performance gap between commercial and research dialogue systems.

Appendix A

Trace of FADE in the SmartWeb System

In the following, we will provide a simplified processing trace of FADE in the SMARTWEB system. The trace is organized as follows: For each user utterance, we will first provide the processing trace of the interpretation phase including the trace for the integration of the selected hypothesis into the discourse context. Then, we will show the system response and finally the processing trace of FADE for integrating the system turn into the discourse context. Note that due to layout restrictions the namespace information “smartweb.semanticweb.org” is replaced by ...; these ellipses also indicate omitted closing XML-tags. Moreover, to keep the trace short and readable, FADE has been run in a configuration with deactivated spreading activation.

User: *“Zeige mir italienische Restaurants in Saarbrücken.”*

```
ResolveMultipleHypotheses(): processing 1 interpretation hypotheses
ResolveMultipleHypotheses(): looping over hypotheses
  about to resolve temporal references
  hypothesis does not contain any temporal references...
Processing hypothesis: zeige mir italienische Restaurants in Saarbru'cken
-> ResolveReferringExpressions(): about to resolve referring expressions
  this hypothesis contains 0 referring expressions
Computing score:
nlu score is = 0.6667
npr = 1.0
dpg = 1.0
udr = 1.0
Score of the current hypothesis: 0.916675
ResolveMultipleHypotheses(): finished; score of winning hypothesis is: 0.916675
Content of wining hypothesis: <object type="discourse#Query" id="http://.../individuals/fade#f3">
...
<slot name="discourse#focus">
<object type="discourse#Focus" id="http://.../individuals/fade#f11">
  <slot name="discourse#qeVariable">
    <object type="discourse#QEVariable" id="http://.../individuals/fade#f19">
      <slot name="discourse#varContext">
        <object type="discourse#VarContext" id="http://.../individuals/fade#f20">
          <slot name="discourse#contextObject">
            <object type="smartsumo#Map" id="http://.../individuals/fade#f5">
              <slot name="smartsumo#POI">
                <object type="navigation#ItalianRestaurant" id="http://.../individuals/fade#f9"/>
              </slot>
            <slot name="smartsumo#centerAddress">
              <object type="smartsumo#Address" id="http://.../individuals/fade#f6">
```

```

    <slot name="smartsumo#hasCity">
      <object type="smartsumo#City" id="http://.../individuals/fade#f7">
        <slot name="smartdolce#HAS-DENOMINATION">
          <object type="smartdolce#denomination" id="http://.../individuals/fade#f8">
            <slot name="smartdolce#NAME">
              <value type="String">Saarbru'cken</value>
            </slot>
          </object>
        </slot>
      </object>
    </slot>
  <slot name="discourse#text">
    <value type="String">zeige mir italienische Restaurants in Saarbru'cken</value>
  </slot>
  <slot name="discourse#content">
    <object type="discourse#QEPattern" id="http://.../individuals/fade#f4">
      <slot name="discourse#patternArg">
        <object type="smartsumo#Map" id="http://.../individuals/fade#f5"/>
      </slot>
    </object>
  </slot>
</object>

```

Action of type output with variable name interpretation by rule outputResult(withoutAttentionalStatus)
 adding a new user turn to the discourse context
 updating global focus
 updating local focus: current turn is a user turn
 found 1 referring expressions.
 found 1 discourseObjects
 integrating 1 DiscourseObject(s)
 - adding new DO [navigation#ItalianRestaurant(http://.../individuals/fade#f33)]
 Current configuration of the Referential Layer:
 [navigation#ItalianRestaurant(http://.../individuals/fade#f33)] d0 activation is 1.0

System: "Karte mit italienischen Restaurants."



adding a new system turn to the discourse context
 updating global focus
 updating local focus: current turn is a system turn
 integrating 1 DiscourseObject(s)
 - adding new DO [smartsumo#Map(http://.../individuals/goal#i22061)]
 Current configuration of the Referential Layer:
 [smartsumo#Map(http://.../individuals/goal#i22061)] d0 activation is 1.0
 [navigation#ItalianRestaurant(http://.../individuals/fade#f33)] d0 activation is 0.8
 Received an update of the configuration of the Physical Environment
 looks like a MPEG7-based representation encompassing 12 still regions
 about to analyse spatial organization...
 standard derivation is 0.9574271077563381
 [Name: coordinates are 123 168, Name: coordinates are 124 177,

```

    Name: coordinates are 181 184, Name: coordinates are 189 119]
  [Name: coordinates are 50 93, Name: coordinates are 58 36,
  Name: coordinates are 66 24, Name: coordinates are 73 51,
  Name: coordinates are 94 83, Name: coordinates are 98 91,
  Name: coordinates are 145 97, Name: coordinates are 150 76]
  looks like a two column list
Action of type output with variable name output by rule handleUIDStatus
addToOutputQueue(): received wme! WME8 [UpdatePhysicalContext\1]
  integrating 1 DiscourseObject(s)
    - adding new DO [smartsumo#Map(http://.../individuals/goal#i22061)]
  Current configuration of the Referential Layer:
    [smartsumo#Map(http://.../individuals/goal#i22061)] d0 activation is 0.8
    [navigation#ItalianRestaurant(http://.../individuals/fade#f33)] d0 activation is 0.60
update physical context finished!

```

User: *"Gib mir mehr Informationen dazu."* [pointing gesture]

```

Action of type output with variable name goal by rule handlePointingGesture(2)
wme! WME-1 [discourse#PointingGesture\1]
  processing pointing gesture
got content of the pointing gesture
  integrating 1 DiscourseObject(s)
    - adding new DO [discourse#PointingGesture()]
  Current configuration of the Referential Layer:
    [discourse#PointingGesture()] d0 activation is 1.0
    [smartsumo#Map(http://.../individuals/goal#i22061)] d0 activation is 0.8
    [navigation#ItalianRestaurant(http://.../individuals/fade#f33)] d0 activation is 0.60
Action of type output with variable name goal by rule handlePointingGesture(2)
addToOutputQueue(): received wme! WME-1 [discourse#PointingGesture\1]
  processing pointing gesture
got content of the pointing gesture
  integrating 1 DiscourseObject(s)
    - adding new DO [navigation#ItalianRestaurant(http://.../individuals/goal#i22152)]
  Current configuration of the Referential Layer:
    [navigation#ItalianRestaurant(http://.../individuals/goal#i22152)] d0 activation is 1.0
    [discourse#PointingGesture()] d0 activation is 1.0
    [smartsumo#Map(http://.../individuals/goal#i22061)] d0 activation is 0.8
    [navigation#ItalianRestaurant(http://.../individuals/fade#f33)] d0 activation is 0.60
ResolveMultipleHypotheses(): processing 4 interpretation hypotheses
ResolveMultipleHypotheses(): looping over hypotheses
  about to resolve temporal references
  hypothesis does not contain any temporal references...
Processing hypothesis: gib mir mehr Informationen dazu
-> ResolveReferringExpressions(): about to resolve referring expressions
  this hypothesis contains 1 referring expressions
  about to resolve reference [smartsumo#Movie(http://.../individuals/fade#f72)]
  first trying hypotheses that this is a reference to the physical context
-> resolveReferenceToTwoColumnList()
  hypothesis was wrong: this is not a reference to a physical object
  performing reference resolution
  looping over 4 Referential objects
  current DO is of type [navigation#ItalianRestaurant(http://.../individuals/goal#i22152)]
  activation is 1.0
  trying to treat it as a deictic reference
resolveDeicticExpression (number of sponsors 1)
found potential sponsor: [navigation#ItalianRestaurant(http://.../individuals/goal#i22152)]
type is pointingGesture number
unifying [smartsumo#Movie(http://.../individuals/fade#f72)] and
[navigation#ItalianRestaurant(http://.../individuals/goal#i22152)]
this is not a deictic reference; trying to treat it as an anaphoric reference
resolveDeicticExpressionToFocusedObject
this is not an anaphoric reference; trying to treat it as a partial reference
trying to treat it as a total discourse anaphoric reference
current DO is of type [discourse#PointingGesture()] activation is 1.0
trying to treat it as a deictic reference
resolveDeicticExpression (number of sponsors 1)
found potential sponsor: [discourse#PointingGesture()] type is pointingGesture number

```

```

unifying [smartsumo#Movie(http://.../individuals/fade#f72)] and [discourse#PointingGesture()]
  this is not a deictic reference; trying to treat it as an anaphoric reference
resolveDeicticExpressionToFocusedObject
  this is not an anaphoric reference; trying to treat it as a partial reference
  trying to treat it as a total discourse anaphoric reference
  current DO is of type [smartsumo#Map(http://.../individuals/goal#i22061)]
  activation is 0.8
  trying to treat it as a deictic reference
resolveDeicticExpression (number of sponsors 2
  this is not a deictic reference; trying to treat it as an anaphoric reference
resolveDeicticExpressionToFocusedObject
  found potential sponsor: [smartsumo#Map(http://.../individuals/goal#i22061)]
  this is not an anaphoric reference; trying to treat it as a partial reference
  trying to treat it as a total discourse anaphoric reference
  current DO is of type [navigation#ItalianRestaurant(http://.../individuals/fade#f33)]
  activation is 0.60
  trying to treat it as a deictic reference
resolveDeicticExpression (number of sponsors 1
  this is not a deictic reference; trying to treat it as an anaphoric reference
resolveDeicticExpressionToFocusedObject
  this is not an anaphoric reference; trying to treat it as a partial reference
  trying to treat it as a total discourse anaphoric reference
  Computing score:
  nlu score is = 0.6667
  npr = 1.0
  dpq = 0.0
  udr = 1.0
  Score of the current hypothesis: 0.666675
  about to resolve temporal references
  hypothesis does not contain any temporal references...
Processing hypothesis: gib mir mehr Informationen dazu
-> ResolveReferringExpressions(): about to resolve referring expressions
  this hypothesis contains 1 referring expressions
  about to resolve reference [smartsumo#Event(http://.../individuals/fade#f85)]
  first trying hypotheses that this is a reference to the physical context
-> resolveReferenceToTwoColumnList()
  hypothesis was wrong: this is not a reference to a physical object
  performing reference resolution
  looping over 4 Referential objects
  current DO is of type [navigation#ItalianRestaurant(http://.../individuals/goal#i22152)]
  activation is 1.0
  trying to treat it as a deictic reference
resolveDeicticExpression (number of sponsors 1
  found potential sponsor: [navigation#ItalianRestaurant(http://.../individuals/goal#i22152)]
  type is pointingGesture number
unifying [smartsumo#Event(http://.../individuals/fade#f85)] and
  [navigation#ItalianRestaurant(http://.../individuals/goal#i22152)]
  this is not a deictic reference; trying to treat it as an anaphoric reference
resolveDeicticExpressionToFocusedObject
  this is not an anaphoric reference; trying to treat it as a partial reference
  trying to treat it as a total discourse anaphoric reference
  current DO is of type [discourse#PointingGesture()] activation is 1.0
  trying to treat it as a deictic reference
resolveDeicticExpression (number of sponsors 1
  found potential sponsor: [discourse#PointingGesture()] type is pointingGesture number
unifying [smartsumo#Event(http://.../individuals/fade#f85)] and [discourse#PointingGesture()]
  this is not a deictic reference; trying to treat it as an anaphoric reference
resolveDeicticExpressionToFocusedObject
  this is not an anaphoric reference; trying to treat it as a partial reference
  trying to treat it as a total discourse anaphoric reference
  current DO is of type [smartsumo#Map(http://.../individuals/goal#i22061)]
  activation is 0.8
  trying to treat it as a deictic reference
resolveDeicticExpression (number of sponsors 2
  this is not a deictic reference; trying to treat it as an anaphoric reference
resolveDeicticExpressionToFocusedObject

```



```

found potential sponsor: [smartsumo#Map(http://.../individuals/goal#i22061)]
  this is not an anaphoric reference; trying to treat it as a partial reference
  trying to treat it as a total discourse anaphoric reference
  current DO is of type [navigation#ItalianRestaurant(http://.../individuals/fade#f33)]
  activation is 0.60
  trying to treat it as a deictic reference
resolveDeicticExpression (number of sponsors 1
  this is not a deictic reference; trying to treat it as a anaphoric reference
resolveDeicticExpressionToFocusedObject
  this is not an anaphoric reference; trying to treat it as a partial reference
  trying to treat it as a total discourse anaphoric reference
Computing score:
nlu score is = 0.6667
npr = 1.0
dpg = 0.0
udr = 1.0
Score of the current hypothesis: 0.666675
  about to resolve temporal references
  hypothesis does not contain any temporal references...
Processing hypothesis: gib mir mehr Informationen dazu
-> ResolveReferringExpressions(): about to resolve referring expressions
  this hypothesis contains 1 referring expressions
  about to resolve reference [sportevent#FootballPlayer(http://.../individuals/fade#f99)]
  first trying hypotheses that this is a reference to the physical context
-> resolveReferenceToTwoColumnList()
  hypothesis was wrong: this is not a reference to a physical object
  performing reference resolution
  looping over 4 Referential objects
  current DO is of type [navigation#ItalianRestaurant(http://.../individuals/goal#i22152)]
  activation is 1.0
  trying to treat it as a deictic reference
resolveDeicticExpression (number of sponsors 1
found potential sponsor: [navigation#ItalianRestaurant(http://.../individuals/goal#i22152)]
  type is pointingGesture number
  unifying
  [sportevent#FootballPlayer(http://.../individuals/fade#f99)] and
  [navigation#ItalianRestaurant(http://.../individuals/goal#i22152)]
  this is not a deictic reference; trying to treat it as a anaphoric reference
resolveDeicticExpressionToFocusedObject
  this is not an anaphoric reference; trying to treat it as a partial reference
  trying to treat it as a total discourse anaphoric reference
  current DO is of type [discourse#PointingGesture()] activation is 1.0
  trying to treat it as a deictic reference
resolveDeicticExpression (number of sponsors 1
  found potential sponsor: [discourse#PointingGesture()] type is pointingGesture number
  unifying [sportevent#FootballPlayer(http://.../individuals/fade#f99)] and [discourse#PointingGesture()]
  this is not a deictic reference; trying to treat it as a anaphoric reference
resolveDeicticExpressionToFocusedObject
  this is not an anaphoric reference; trying to treat it as a partial reference
  trying to treat it as a total discourse anaphoric reference
  current DO is of type [smartsumo#Map(http://.../individuals/goal#i22061)]
  activation is 0.8
  trying to treat it as a deictic reference
resolveDeicticExpression (number of sponsors 2
  this is not a deictic reference; trying to treat it as a anaphoric reference
resolveDeicticExpressionToFocusedObject
  found potential sponsor: [smartsumo#Map(http://.../individuals/goal#i22061)]
  this is not an anaphoric reference; trying to treat it as a partial reference
  trying to treat it as a total discourse anaphoric reference
  current DO is of type [navigation#ItalianRestaurant(http://.../individuals/fade#f33)]
  activation is 0.60
  trying to treat it as a deictic reference
resolveDeicticExpression (number of sponsors 1
  this is not a deictic reference; trying to treat it as a anaphoric reference
resolveDeicticExpressionToFocusedObject
  this is not an anaphoric reference; trying to treat it as a partial reference

```

```

    trying to treat it as a total discourse anaphoric reference
Computing score:
nlu score is = 0.6667
npr = 1.0
dpg = 0.0
udr = 1.0
Score of the current hypothesis: 0.666675
    about to resolve temporal references
    hypothesis does not contain any temporal references...
Processing hypothesis: gib mir mehr Informationen dazu
-> ResolveReferringExpressions(): about to resolve referring expressions
    this hypothesis contains 1 referring expressions
    about to resolve reference [smartsumo#PublicBuilding(http://.../individuals/fade#f109)]
    first trying hypotheses that this is a reference to the physical context
-> resolveReferenceToTwoColumnList()
    hypothesis was wrong: this is not a reference to a physical object
    performing reference resolution
    looping over 4 Referential objects
    current DO is of type [navigation#ItalianRestaurant(http://.../individuals/goal#i22152)]
    activation is 1.0
    trying to treat it as a deictic reference
resolveDeicticExpression (number of sponsors 1
found potential sponsor: [navigation#ItalianRestaurant(http://.../individuals/goal#i22152)]
    type is pointingGesture number
    unifying [smartsumo#PublicBuilding(http://.../individuals/fade#f109)] and
    [navigation#ItalianRestaurant(http://.../individuals/goal#i22152)]
unification: ok
checking id mapping for id http://.../individuals/goal#i22152
checking id mapping for id http://.../individuals/goal#i22152
id mapping applied
    got a result from resolveDeicticExpression()
got 1 focus objects
got focused object <object type="smartsumo#PublicBuilding" id="http://.../individuals/fade#f109">
  <slot name="linginfo#lingInfo">
    <object type="discourse#RefProp" id="http://.../individuals/fade#f110">
      <slot name="discourse#type">
        <value type="String">def</value>
      </slot>
    </object>
  </slot>
</object>
looks like this focus object was embedded in the refProp
replaced the object
Computing score:
nlu score is = 0.6667
npr = 1.0
dpg = 0.5
udr = 1.0
Score of the current hypothesis: 0.791675
ResolveMultipleHypotheses(): finished; score of winning hypothesis is: 0.791675
Content of wining hypothesis: <object type="discourse#Query" id="http://.../individuals/fade#f107">
...
<slot name="discourse#focus">
  <object type="discourse#Focus" id="http://.../individuals/fade#f112">
    <slot name="discourse#qeVariable">
      <object type="discourse#QEVARIABLE" id="http://.../individuals/fade#f117">
        <slot name="discourse#varContext">
          <object type="discourse#VarContext" id="http://.../individuals/fade#f118">
            <slot name="discourse#contextObject">
              <object type="navigation#ItalianRestaurant" id="http://.../individuals/goal#i22152">
                <slot name="smartdolce#HAS-DENOMINATION">
                  <object type="smartdolce#denomination" id="http://.../individuals/goal#i22153">
                    <slot name="smartdolce#NAME">
                      <value type="String">Pizzeria La Trattoria</value>
                    </slot>
                  </object>
                </slot>
              </object>
            </slot>
          </object>
        </slot>
      </object>
    </slot>
  </object>
</slot>

```

```

...
</slot>
<slot name="discourse#text"> <value type="String">gib mir mehr Informationen dazu</value> </slot>
<slot name="discourse#content">
  <object type="discourse#QEPattern" id="http://.../individuals/fade#f108">
    <slot name="discourse#patternArg">
      <object type="navigation#ItalianRestaurant" id="http://.../individuals/goal#i22152">
        <slot name="smartdolce#HAS-DENOMINATION">
          <object type="smartdolce#denomination" id="http://.../individuals/goal#i22153"/>
        </slot>
      </object>
    </slot>
  </object>
  ...
</object>

```

Action of type output with variable name interpretation by rule outputResult(withoutAttentionalStatus)
adding a new user turn to the discourse context
updating global focus
updating local focus: current turn is a user turn
found 1 referring expressions.
found 1 discourseObjects
integrating 1 DiscourseObject(s)
- adding new DO [navigation#ItalianRestaurant(http://.../individuals/fade#f127)]
Current configuration of the Referential Layer:
[navigation#ItalianRestaurant(http://.../individuals/fade#f127)] d0 activation is 1.0
[navigation#ItalianRestaurant(http://.../individuals/goal#i22152)] d0 activation is 0.8
[discourse#PointingGesture()] d0 activation is 0.8
[smartsumo#Map(http://.../individuals/goal#i22061)] d0 activation is 0.60
[navigation#ItalianRestaurant(http://.../individuals/fade#f33)] d0 activation is 0.40

System: *"Pizzeria La Trattoria."*



adding a new system turn to the discourse context
updating global focus
updating local focus: current turn is a system turn
integrating 1 DiscourseObject(s)
- adding new DO [navigation#ItalianRestaurant(http://.../individuals/goal#i22524)]
Current configuration of the Referential Layer:

[navigation#ItalianRestaurant(http://.../individuals/goal#i22524)] d0 activation is 1.0
[navigation#ItalianRestaurant(http://.../individuals/fade#f127)] d0 activation is 0.8
[navigation#ItalianRestaurant(http://.../individuals/goal#i22152)] d0 activation is 0.60
[discourse#PointingGesture()] d0 activation is 0.60
[smartsumo#Map(http://.../individuals/goal#i22061)] d0 activation is 0.40
[navigation#ItalianRestaurant(http://.../individuals/fade#f33)] d0 activation is 0.20

User: *"Wie komme ich dahin?"*

```

ResolveMultipleHypotheses(): processing 1 interpretation hypotheses
ResolveMultipleHypotheses(): looping over hypotheses
  about to resolve temporal references
  hypothesis does not contain any temporal references...
Processing hypothesis: wie komme ich dahin
  -> ResolveReferringExpressions(): about to resolve referring expressions
  this hypothesis contains 1 referring expressions
  about to resolve reference [smartsumo#Address(http://.../individuals/fade#f163)]
  first trying hypotheses that this is a reference to the physical context
-> resolveReferenceToTwoColumnList()
  hypothesis was wrong: this is not a reference to a physical object
  performing reference resolution
  looping over 6 Referential objects
  current DO is of type [navigation#ItalianRestaurant(http://.../individuals/goal#i22524)]
  activation is 1.0
  trying to treat it as a deictic reference
resolveDeicticExpression (number of sponsors 1)
  this is not a deictic reference; trying to treat it as anaphoric reference
resolveDeicticExpressionToFocusedObject
  this is not an anaphoric reference; trying to treat it as a partial reference
  got result from resolvePartialAnaphora()
got 1 focus objects
got focused object <object type="navigation#CarDriving" id="http://.../individuals/fade#f162">
  <slot name="smartsumo#hasGoal">
    <object type="smartsumo#Address" id="http://.../individuals/fade#f163">
      <slot name="linginfo#lingInfo">
        <object type="discourse#RefProp" id="http://.../individuals/fade#f164">
          <slot name="discourse#type">
            <value type="String">def</value>
          </slot>
        </object>
      </slot>
    </object>
  </slot>
</object>
</slot>
</object>
Computing score:
nlu score is = 0.75
npr = 1.0
dpg = 1.0
udr = 1.0
Score of the current hypothesis: 0.9375
ResolveMultipleHypotheses(): finished; score of winning hypothesis is: 0.9375
Content of wining hypothesis: <object type="discourse#Query" id="http://.../individuals/fade#f160">
  ...
  <slot name="discourse#focus">
    <object type="discourse#Focus" id="http://.../individuals/fade#f166">
      <slot name="discourse#qeVariable">
        <object type="discourse#QEVariable" id="http://.../individuals/fade#f169">
          <slot name="discourse#varContext">
            <object type="discourse#VarContext" id="http://.../individuals/fade#f170">
              <slot name="discourse#contextObject">
                <object type="navigation#CarDriving" id="http://.../individuals/fade#f162">
                  <slot name="smartsumo#hasGoal">
                    <object type="smartsumo#Address" id="http://.../individuals/goal#i22532">
                      <slot name="linginfo#lingInfo">
                        <object type="discourse#RefProp" id="http://.../individuals/fade#f164">
                          <slot name="discourse#type">
                            <value type="String">def</value>
                          </slot>
                        </object>
                      </slot>
                    </object>
                  </slot>
                </object>
              </slot>
            </object>
          </slot>
        </object>
      </slot>
    </object>
  </slot>
  <slot name="smartsumo#GEOPOSITION">
    ...
  </slot>
  <slot name="smartsumo#POSTALCODE">
    <value type="String">66111</value>
  </slot>

```

```

        </slot>
        <slot name="smartsumo#ROADNAME">
          <value type="String">Nauwieserplatz</value>
        </slot>
        <slot name="smartsumo#HOUSENUMBER">
          <value type="String">3</value>
        </slot>
        <slot name="smartsumo#hasCity">
          ...
        </slot>
      </object>
    ...
  </slot>
  <slot name="discourse#text"> <value type="String">wie komme ich dahin</value> </slot>
  <slot name="discourse#content">
    <object type="discourse#QEPattern" id="http://.../individuals/fade#f161">
      <slot name="discourse#patternArg">
        <object type="navigation#CarDriving" id="http://.../individuals/fade#f162"/>
      </slot>
    </object>
  </slot>
</object>

```

Action of type output with variable name interpretation by rule outputResult(withoutAttentionalStatus)
 adding a new user turn to the discourse context
 updating global focus
 updating local focus: current turn is a user turn
 found 0 referring expressions.
 found 0 discourseObjects
 integrating 0 DiscourseObject(s)
 Current configuration of the Referential Layer:
 [navigation#ItalianRestaurant(http://.../individuals/goal#i22524)] d0 activation is 0.8
 [navigation#ItalianRestaurant(http://.../individuals/fade#f127)] d0 activation is 0.60
 [navigation#ItalianRestaurant(http://.../individuals/goal#i22152)] d0 activation is 0.40
 [discourse#PointingGesture()] d0 activation is 0.40
 [smartsumo#Map(http://.../individuals/goal#i22061)] d0 activation is 0.20

System: *“Die berechnete Route wird gezeigt.”*



adding a new system turn to the discourse context
 updating global focus
 updating local focus: current turn is a system turn
 integrating 1 DiscourseObject(s)
 - adding new DO [navigation#CarDriving(http://.../individuals/goal#i23331)]
 Current configuration of the Referential Layer:
 [navigation#CarDriving(http://.../individuals/goal#i23331)] d0 activation is 1.0

```
[navigation#ItalianRestaurant(http://.../individuals/goal#i22524)] d0 activation is 0.60
[navigation#ItalianRestaurant(http://.../individuals/fade#f127)] d0 activation is 0.40
[navigation#ItalianRestaurant(http://.../individuals/goal#i22152)] d0 activation is 0.20
[discourse#PointingGesture()] d0 activation is 0.20
```

User: "Wie wird morgen das Wetter?"

```
ResolveMultipleHypotheses(): processing 2 interpretation hypotheses
ResolveMultipleHypotheses(): looping over hypotheses
  about to resolve temporal references
    found 1 SAtimepoint(s)
      timeRequest is tomorrow
      resolved temporal expression is:
      time-interval: 17.02.2007 0:00 - 23:59
Processing hypothesis: wie wird morgen das Wetter
  -> ResolveReferringExpressions(): about to resolve referring expressions
    this hypothesis contains 0 referring expressions
Computing score:
nlu score is = 0.6667
npr = 1.0
dpg = 1.0
udr = 1.0
Score of the current hypothesis: 0.916675
  about to resolve temporal references
    found 1 SAtimepoint(s)
      resolved temporal expression is:
      time-interval: 17.02.2007 0:00 - 23:59
Processing hypothesis: wie wird morgen das Wetter
  -> ResolveReferringExpressions(): about to resolve referring expressions
    this hypothesis contains 0 referring expressions
Computing score:
nlu score is = 0.6667
npr = 1.0
dpg = 1.0
udr = 1.0
Score of the current hypothesis: 0.916675
ResolveMultipleHypotheses(): finished; score of winning hypothesis is: 0.916675
Content of wining hypothesis: <object type="discourse#Query" id="http://.../individuals/fade#f204">
...
  <slot name="discourse#focus">
    <object type="discourse#Focus" id="http://.../individuals/fade#f210">
      <slot name="discourse#qeVariable">
        <object type="discourse#QEVariable" id="http://.../individuals/fade#f213">
          <slot name="discourse#varContext">
            <object type="discourse#VarContext" id="http://.../individuals/fade#f214">
              <slot name="discourse#contextObject">
                <object type="smartsumo#weather-forecast" id="http://.../individuals/fade#f206">
                  <slot name="smartdolce#PRESENT-AT">
                    <object type="smartdolce#time-interval">
                      <slot name="smartdolce#ENDS">
                        <object type="smartdolce#time-point">
                          <slot name="smartdolce#DAY"> <value type="String">17</value> </slot>
                          <slot name="smartdolce#YEAR"> <value type="String">2007</value> </slot>
                          <slot name="smartdolce#HOUR"> <value type="String">23</value> </slot>
                          <slot name="smartdolce#MINUTE"> <value type="String">59</value> </slot>
                          <slot name="smartdolce#MONTH"> <value type="String">2</value> </slot>
                        </object>
                      </slot>
                    </object>
                  </slot>
                </object>
              </slot>
            </object>
          </slot>
        </object>
      </slot>
    </object>
  </slot>
  <slot name="smartdolce#BEGINS">
    <object type="smartdolce#time-point">
      <slot name="smartdolce#DAY"> <value type="String">17</value> </slot>
      <slot name="smartdolce#YEAR"> <value type="String">2007</value> </slot>
      <slot name="smartdolce#HOUR"> <value type="String">0</value> </slot>
      <slot name="smartdolce#MINUTE"> <value type="String">00</value> </slot>
      <slot name="smartdolce#MONTH"> <value type="String">2</value> </slot>
    </object>
  </slot>
  ...
```

```

</slot>
<slot name="discourse#text">
  <value type="String">wie wird morgen das Wetter</value>
</slot>
<slot name="discourse#content">
  <object type="discourse#QEPattern" id="http://.../individuals/fade#f205">
    <slot name="discourse#patternArg">
      <object type="smartsumo#weather-forecast" id="http://.../individuals/fade#f206"/>
    </slot>
  </object>
</slot>
</object>

```

Action of type output with variable name interpretation by rule outputResult(withoutAttentionalStatus)
 adding a new user turn to the discourse context
 updating global focus
 updating local focus: current turn is a user turn
 found 0 referring expressions.
 found 0 discourseObjects
 integrating 0 DiscourseObject(s)
 Current configuration of the Referential Layer:
 [navigation#CarDriving(http://.../individuals/goal#i23331)] d0 activation is 0.8
 [navigation#ItalianRestaurant(http://.../individuals/goal#i22524)] d0 activation is 0.40
 [navigation#ItalianRestaurant(http://.../individuals/fade#f127)] d0 activation is 0.20

System: "Wettervorhersage für Saarbrücken."



adding a new system turn to the discourse context
 updating global focus
 updating local focus: current turn is a system turn
 integrating 1 DiscourseObject(s)
 - adding new D0 [smartsumo#weather-forecast(http://.../individuals/goal#i23791)]
 Current configuration of the Referential Layer:
 [smartsumo#weather-forecast(http://.../individuals/goal#i23791)] d0 activation is 1.0
 [navigation#CarDriving(http://.../individuals/goal#i23331)] d0 activation is 0.60
 [navigation#ItalianRestaurant(http://.../individuals/goal#i22524)] d0 activation is 0.20

User: "Und übermorgen?"

ResolveMultipleHypotheses(): processing 1 interpretation hypotheses
 ResolveMultipleHypotheses(): looping over hypotheses
 about to resolve temporal references
 found 1 SATimepoint(s)
 resolved temporal expression is:
 time-interval: 18.02.2007 18:00 - 23:59
 Processing hypothesis: und u'bermorgen
 hypothesis contains partial - potentially elliptical - content

```

-> successfully integrated the fragment into the previous user utterance
-> ResolveReferringExpressions(): about to resolve referring expressions
    this hypothesis contains 0 referring expressions
Computing score:
nlu score is = 0.3333333333333333
npr = 1.0
dpg = 1.0
udr = 1.0
Score of the current hypothesis: 0.8333333333333333
trying to resolve elliptical expression; using the previous system turn...
about to resolve ellipsis to a previous user turn
resolved elliptical expression
ResolveMultipleHypotheses(): finished; score of winning hypothesis is: 0.8333333333333333
Content of wining hypothesis: <object type="discourse#Query" id="http://.../individuals/fade#f236">
...
<slot name="discourse#focus">
  <object type="discourse#Focus" id="http://.../individuals/fade#f255">
    <slot name="discourse#qeVariable">
      <object type="discourse#QEVariable" id="http://.../individuals/fade#f257">
        <slot name="discourse#varContext">
          <object type="discourse#VarContext" id="http://.../individuals/fade#f258">
            <slot name="discourse#contextObject">
              <object type="smartsumo#weather-forecast" id="http://.../individuals/fade#f238">
                <slot name="smartdolce#GENERIC-LOCATION">
                  <object type="smartsumo#Address" id="http://.../individuals/fade#f239">
                    ...
                  </object>
                </slot>
              <slot name="smartdolce#PRESENT-AT">
                <object type="smartdolce#time-interval" id="http://.../individuals/fade#f243">
                  ...
                </object>
              </slot>
            </slot>
          </slot>
        </slot>
      </slot>
    </slot>
  </slot>
<slot name="discourse#text">
  <value type="String">und u'bermorgen</value>
</slot>
<slot name="discourse#content">
  <object type="discourse#QEPattern" id="http://.../individuals/fade#f237">
    <slot name="discourse#patternArg">
      <object type="smartsumo#weather-forecast" id="http://.../individuals/fade#f238"/>
    </slot>
  </object>
</slot>
<slot name="discourse#hasDiscourseOrigin">
  <value type="String">Ellipse</value>
</slot>
</object>
Action of type output with variable name interpretation by rule outputResult(withoutAttentionalStatus)
adding a new user turn to the discourse context
  updating global focus
  updating local focus: current turn is a user turn
  found 0 referring expressions.
found 0 discourseObjects
  integrating 0 DiscourseObject(s)
Current configuration of the Referential Layer:
  [smartsumo#weather-forecast(http://.../individuals/goal#i23791)] d0 activation is 0.8
  [navigation#CarDriving(http://.../individuals/goal#i23331)] d0 activation is 0.40

```

System: "Wettervorhersage für Saarbrücken."



```

adding a system turn to the discourse context
  updating global focus
  updating local focus: current turn is a system turn
    integrating 1 DiscourseObject(s)
      - adding new DO [smartsumo#weather-forecast(http://.../individuals/goal#i24202)]
Current configuration of the Referential Layer:
  [smartsumo#weather-forecast(http://.../individuals/goal#i24202)] d0 activation is 1.0
  [smartsumo#weather-forecast(http://.../individuals/goal#i23791)] d0 activation is 0.60
  [navigation#CarDriving(http://.../individuals/goal#i23331)] d0 activation is 0.20

```

User: *“Was ist heute Abend los?”*

```

ResolveMultipleHypotheses(): processing 1 interpretation hypotheses
ResolveMultipleHypotheses(): looping over hypotheses
  about to resolve temporal references
    found 1 SATimepoint(s)
      timeRequest is today
      resolved temporal expression is:
        time-interval: 16.02.2007 18:00 - 23:59
Processing hypothesis: was ist heute abend los
  -> ResolveReferringExpressions(): about to resolve referring expressions
    this hypothesis contains 0 referring expressions
Computing score:
  nlu score is = 0.6667
  npr = 1.0
  dpq = 1.0
  udr = 1.0
  Score of the current hypothesis: 0.916675
ResolveMultipleHypotheses(): finished; score of winning hypothesis is: 0.916675
Content of winning hypothesis: <object type="discourse#Query" id="http://.../individuals/fade#f309">
...
<slot name="discourse#focus">
  <object type="discourse#Focus" id="http://.../individuals/fade#f316">
    <slot name="discourse#qeVariable">
      <object type="discourse#QEVariable" id="http://.../individuals/fade#f322">
        <slot name="discourse#varContext">
          <object type="discourse#VarContext" id="http://.../individuals/fade#f323">
            <slot name="discourse#contextObject">
              <object type="smartsumo#Event" id="http://.../individuals/fade#f311">
                <slot name="smartdolce#HAPPENS-AT">
                  <object type="smartdolce#time-interval">
                    ...
                  </object>
                </slot>
              </object>
            </slot>
          </object>
        </slot>
      </object>
    </slot>
  </object>
</slot>

```

```

<slot name="discourse#text">
  <value type="String">was ist heute abend los</value>
</slot>
<slot name="discourse#content">
  <object type="discourse#QEPattern" id="http://.../individuals/fade#f310">
    <slot name="discourse#patternArg">
      <object type="smartsumo#Event" id="http://.../individuals/fade#f311"/>
    </slot>
  </object>
</slot>
</object>
Action of type output with variable name interpretation by rule outputResult(withoutAttentionalStatus)
adding a new user turn to the discourse context
  updating global focus
  updating local focus: current turn is a user turn
  found 1 referring expressions.
found 1 discourseObjects
  integrating 1 DiscourseObject(s)
  - adding new DO [smartsumo#StationaryArtifact(http://.../individuals/fade#f336)]
Current configuration of the Referential Layer:
[smartsumo#StationaryArtifact(http://.../individuals/fade#f336)] d0 activation is 1.0
[smartsumo#weather-forecast(http://.../individuals/goal#i24202)] d0 activation is 0.8
[smartsumo#weather-forecast(http://.../individuals/goal#i23791)] d0 activation is 0.40

```

System: "5 Veranstaltungen."



```

adding a new system turn to the discourse context
  updating global focus
  updating local focus: current turn is a system turn

```

User: "Wo findet diese [pointing gesture] Veranstaltung statt?"

```

Action of type output with variable name goal by rule handlePointingGesture(2)
addToOutputQueue(): received wme! WME-1 [discourse#PointingGesture\1]
  processing pointing gesture
got content of the pointing gesture
  integrating 1 DiscourseObject(s)
  - adding new DO [smartsumo#Event(http://.../individuals/goal#i25146)]
Current configuration of the Referential Layer:
[smartsumo#Event(http://.../individuals/goal#i25146)] d0 activation is 1.0
[smartsumo#StationaryArtifact(http://.../individuals/fade#f336)] d0 activation is 1.0
[smartsumo#weather-forecast(http://.../individuals/goal#i24202)] d0 activation is 0.8
[smartsumo#weather-forecast(http://.../individuals/goal#i23791)] d0 activation is 0.40
ResolveMultipleHypotheses(): processing 1 interpretation hypotheses
ResolveMultipleHypotheses(): looping over hypotheses
  about to resolve temporal references
  hypothesis does not contain any temporal references...

```

```

Processing hypothesis: wo findet diese Veranstaltung statt
-> ResolveReferringExpressions(): about to resolve referring expressions
    this hypothesis contains 1 referring expressions
    about to resolve reference [smartsumo#Event(http://.../individuals/fade#f371)]
    first trying hypotheses that this is a reference to the physical context
-> resolveReferenceToTwoColumnList()
    hypothesis was wrong: this is not a reference to a physical object
    performing reference resolution
    looping over 4 Referential objects
    current DO is of type [smartsumo#Event(http://.../individuals/goal#i25146)]
    activation is 1.0
    trying to treat it as a deictic reference
resolveDeicticExpression (number of sponsors 1
found potential sponsor: [smartsumo#Event(http://.../individuals/goal#i25146)]
type is pointingGesture number singular
unifying [smartsumo#Event(http://.../individuals/fade#f371)] and
[smartsumo#Event(http://.../individuals/goal#i25146)]
unification: ok
checking id mapping for id http://.../individuals/goal#i25146
checking id mapping for id http://.../individuals/goal#i25146
id mapping applied
    got a result from resolveDeicticExpression()
    got 1 focus objects
    got focused object <object type="smartsumo#StationaryArtifact" id="http://.../individuals/fade#f372"/>
    looks like this focus object was embedded in the refProp
    replaced the object
    Computing score:
    nlu score is = 0.6667
    npr = 1.0
    dpg = 1.0
    udr = 1.0
    Score of the current hypothesis: 0.916675
ResolveMultipleHypotheses(): finished; score of winning hypothesis is: 0.916675
Content of wining hypothesis: <object type="discourse#Query" id="http://.../individuals/fade#f369">
...
<slot name="discourse#focus">
  <object type="discourse#Focus" id="http://.../individuals/fade#f376">
    <slot name="discourse#qeVariable">
      <object type="discourse#QEVariable" id="http://.../individuals/fade#f380">
        <slot name="discourse#varContext">
          <object type="discourse#VarContext" id="http://.../individuals/fade#f381">
            <slot name="discourse#contextObject">
              <object type="smartsumo#StationaryArtifact" id="http://.../individuals/goal#i25151">
                <slot name="smartdolce#HAS-DENOMINATION">
                  <object type="smartdolce#denomination" id="http://.../individuals/goal#i25152">
                    <slot name="smartdolce#NAME">
                      <value type="String">Home</value>
                    </slot>
                  </object>
                </slot>
              </object>
            </slot>
          </object>
        </slot>
      </object>
    </slot>
  </object>
  <slot name="smartsumo#hasAddress">
    ...
  </slot>
</object>
...
</slot>
<slot name="discourse#text">
  <value type="String">wo findet diese Veranstaltung statt</value>
</slot>
<slot name="discourse#content">
  <object type="discourse#QEPattern" id="http://.../individuals/fade#f370">
    <slot name="discourse#patternArg">
      <object type="smartsumo#Event" id="http://.../individuals/goal#i25146">
        ...
      </object>
    </slot>
  </object>
</slot>

```

```

    </object>
  </slot>
</object>
Action of type output with variable name interpretation by rule outputResult(withoutAttentionalStatus)
adding a new user turn to the discourse context
  updating global focus
  updating local focus: current turn is a user turn
    found 1 referring expressions.
found 1 discourseObjects
  integrating 1 DiscourseObject(s)
    - adding new DO [smartsumo#StationaryArtifact(http://.../individuals/fade#f391)]
Current configuration of the Referential Layer:
[smartsumo#StationaryArtifact(http://.../individuals/fade#f391)] d0 activation is 1.0
[smartsumo#Event(http://.../individuals/goal#i25146)] d0 activation is 0.8
[smartsumo#StationaryArtifact(http://.../individuals/fade#f336)] d0 activation is 0.8
[smartsumo#weather-forecast(http://.../individuals/goal#i24202)] d0 activation is 0.60
[smartsumo#weather-forecast(http://.../individuals/goal#i23791)] d0 activation is 0.20

```

System: "Home."



```

adding a new system turn to the discourse context
  updating global focus
  updating local focus: current turn is a system turn
    integrating 1 DiscourseObject(s)
      - adding new DO [smartsumo#StationaryArtifact(http://.../individuals/goal#i25289)]
Current configuration of the Referential Layer:
[smartsumo#StationaryArtifact(http://.../individuals/goal#i25289)] d0 activation is 1.0
[smartsumo#StationaryArtifact(http://.../individuals/fade#f391)] d0 activation is 0.8
[smartsumo#Event(http://.../individuals/goal#i25146)] d0 activation is 0.60
[smartsumo#StationaryArtifact(http://.../individuals/fade#f336)] d0 activation is 0.60
[smartsumo#weather-forecast(http://.../individuals/goal#i24202)] d0 activation is 0.40

```

User: "Wer war 1990 Weltmeister?"

```

ResolveMultipleHypotheses(): processing 1 interpretation hypotheses
ResolveMultipleHypotheses(): looping over hypotheses
  about to resolve temporal references
  hypothesis does not contain any temporal references...
Processing hypothesis: wer war 1990 Weltmeister (confidence: 0.6667)
  -> ResolveReferringExpressions(): about to resolve referring expressions
    this hypothesis contains 0 referring expressions
Computing score:
nlu score is = 0.6667
npr = 1.0
dpg = 1.0
udr = 1.0
Score of the current hypothesis: 0.916675

```

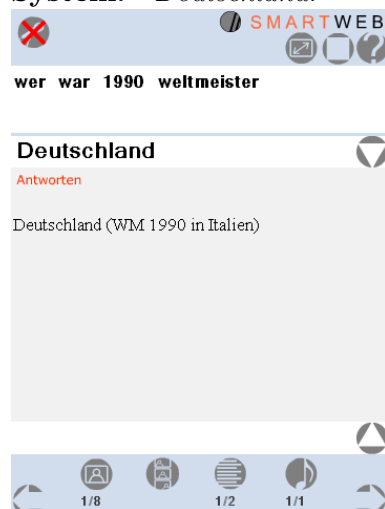
```

ResolveMultipleHypotheses(): finished; score of winning hypothesis is: 0.916675
Content of winning hypothesis: <object type="discourse#Query" id="http://.../individuals/fade#f419">
...
<slot name="discourse#focus">
  <object type="discourse#Focus" id="http://.../individuals/fade#f426">
    <slot name="discourse#qeVariable">
      <object type="discourse#QEVariable" id="http://.../individuals/fade#f428">
        <slot name="discourse#varContext">
          <object type="discourse#VarContext" id="http://.../individuals/fade#f429">
            <slot name="discourse#contextObject">
              <object type="sportevent#FIFAWorldCup" id="http://.../individuals/fade#f421">
                <slot name="smartdolce#HAPPENS-AT">
                  <object type="smartdolce#time-interval" id="http://.../individuals/fade#f423">
                    <slot name="smartdolce#BEGINS">
                      <object type="smartdolce#time-point" id="http://.../individuals/fade#f424">
                        <slot name="smartdolce#YEAR">
                          <value type="String">1990</value>
                        </slot>
                      </object>
                    </slot>
                  </object>
                </slot>
              </object>
            </slot>
          </object>
        </slot>
      </object>
    </slot>
  </object>
  <slot name="sportevent#winner">
    <object type="sportevent#Team" id="http://.../individuals/fade#f422"/>
  </slot>
</slot>
...
<slot name="discourse#text">
  <value type="String">wer war 1990 Weltmeister</value>
</slot>
<slot name="discourse#content">
  <object type="discourse#QEPattern" id="http://.../individuals/fade#f420">
    <slot name="discourse#patternArg">
      <object type="sportevent#FIFAWorldCup" id="http://.../individuals/fade#f421"/>
    </slot>
  </object>
</slot>
</object>

```

Action of type output with variable name interpretation by rule outputResult(withoutAttentionalStatus)

System: "Deutschland."



```

adding a new user turn to the discourse context
updating global focus
updating local focus: current turn is a user turn
found 1 referring expressions.

```

```

found 1 discourseObjects
integrating 1 DiscourseObject(s)
  - adding new DO [sportevent#FIFAWorldCup(http://.../individuals/fade#f451)]
Current configuration of the Referential Layer:
[sportevent#FIFAWorldCup(http://.../individuals/fade#f451)] d0 activation is 1.0
[smartsumo#StationaryArtifact(http://.../individuals/goal#i25289)] d0 activation is 0.8
[smartsumo#StationaryArtifact(http://.../individuals/fade#f391)] d0 activation is 0.60
[smartsumo#Event(http://.../individuals/goal#i25146)] d0 activation is 0.40
[smartsumo#StationaryArtifact(http://.../individuals/fade#f336)] d0 activation is 0.40
[smartsumo#weather-forecast(http://.../individuals/goal#i24202)] d0 activation is 0.20
adding a new system turn to the discourse context
updating global focus
updating local focus: current turn is a system turn
integrating 1 DiscourseObject(s)
  - adding new DO [sportevent#FootballNationalTeam(http://smartweb/kb#WorldCup1990_Team_GER)]
Current configuration of the Referential Layer:
[sportevent#FootballNationalTeam(http://smartweb/kb#WorldCup1990_Team_GER)] d0 activation is 1.0
[sportevent#FIFAWorldCup(http://.../individuals/fade#f451)] d0 activation is 0.8
[smartsumo#StationaryArtifact(http://.../individuals/goal#i25289)] d0 activation is 0.60
[smartsumo#StationaryArtifact(http://.../individuals/fade#f391)] d0 activation is 0.40
[smartsumo#Event(http://.../individuals/goal#i25146)] d0 activation is 0.20
[smartsumo#StationaryArtifact(http://.../individuals/fade#f336)] d0 activation is 0.20

```

User: "Und 2006?"

```

ResolveMultipleHypotheses(): processing 1 interpretation hypotheses
ResolveMultipleHypotheses(): looping over hypotheses
  about to resolve temporal references
  hypothesis does not contain any temporal references...
Processing hypothesis: und 2006
  hypothesis contains partial - potentially elliptical - content
  -> ResolveReferringExpressions(): about to resolve referring expressions
  this hypothesis contains 0 referring expressions
Computing score:
nlu score is = 0.3333333333333333
npr = 1.0
dpg = 1.0
udr = 1.0
Score of the current hypothesis: 0.8333333333333333
trying to resolve elliptical expression; using the previous system turn...
about to resolve ellipsis to a previous user turn
resolved elliptical expression
ResolveMultipleHypotheses(): finished; score of winning hypothesis is: 0.8333333333333333
Content of wining hypothesis: <object type="discourse#Query" id="http://.../individuals/fade#f440">
...
<slot name="discourse#focus">
  <object type="discourse#Focus" id="http://.../individuals/fade#f447">
    <slot name="discourse#qeVariable">
      <object type="discourse#QEVariable" id="http://.../individuals/fade#f449">
        <slot name="discourse#varContext">
          <object type="discourse#VarContext" id="http://.../individuals/fade#f450">
            <slot name="discourse#contextObject">
              <object type="sportevent#FIFAWorldCup" id="http://.../individuals/fade#f451">
                <slot name="linginfo#lingInfo">
                  ...
                </slot>
              <slot name="smartdolce#HAPPENS-AT">
                <object type="smartdolce#time-interval" id="http://.../individuals/fade#f452">
                  <slot name="smartdolce#BEGINS">
                    <object type="smartdolce#time-point" id="http://.../individuals/fade#f453">
                      <slot name="smartdolce#YEAR">
                        <value type="String">2006</value>
                      </slot>
                    </object>
                  </slot>
                </object>
              </slot>
            </object>
          </slot>
        </object>
      </slot>
    </object>
  </slot>

```

```

        <slot name="sportevent#winner">
          <object type="sportevent#Team" id="http://.../individuals/fade#f445"/>
        </slot>
      ...
    </slot>
    <slot name="discourse#text">
      <value type="String">und 2006</value>
    </slot>
    <slot name="discourse#content">
      ...
    </slot>
    <slot name="discourse#hasDiscourseOrigin">
      <value type="String">Ellipse</value>
    </slot>
  </object>

```

Action of type output with variable name interpretation by rule outputResult(withoutAttentionalStatus)
adding a new user turn to the discourse context

updating global focus
updating local focus: current turn is a user turn
found 1 referring expressions.

found 1 discourseObjects

integrating 1 DiscourseObject(s)

- adding new DO [sportevent#FIFAWorldCup(http://.../individuals/fade#f497)]

Current configuration of the Referential Layer:

[sportevent#FIFAWorldCup(http://.../individuals/fade#f497)] dO activation is 1.0

[sportevent#FootballNationalTeam(http://smartweb/kb#WorldCup1990_Team_GER)] dO activation is 0.8

[sportevent#FIFAWorldCup(http://.../individuals/fade#f451)] dO activation is 0.60

[smartsumo#StationaryArtifact(http://.../individuals/goal#i25289)] dO activation is 0.40

[smartsumo#StationaryArtifact(http://.../individuals/fade#f391)] dO activation is 0.20

System: "Italien."



adding a new system turn to the discourse context

updating global focus
updating local focus: current turn is a system turn

integrating 1 DiscourseObject(s)

- adding new DO [sportevent#FootballNationalTeam(http://smartweb/kb#WorldCup2006_Team_ITA)]

Current configuration of the Referential Layer:

[sportevent#FootballNationalTeam(http://smartweb/kb#WorldCup2006_Team_ITA)] dO activation is 1.0

[sportevent#FIFAWorldCup(http://.../individuals/fade#f497)] dO activation is 0.8

[sportevent#FootballNationalTeam(http://smartweb/kb#WorldCup1990_Team_GER)] dO activation is 0.60

[sportevent#FIFAWorldCup(http://.../individuals/fade#f451)] dO activation is 0.40

[smartsumo#StationaryArtifact(http://.../individuals/goal#i25289)] dO activation is 0.20

Received an update of the configuration of the Physical Environment

looks like a MPEG7-based representation encompassing 11 still regions

about to analyse spatial organization...

standard derivation is 0.8528028654224418

```

[Name: Player Andrea Pirlo ITA coordinates are 45 154,
 Name: Player Fabio Cannavaro ITA coordinates are 124 157,
 Name: Player Mauro Camoranesi ITA coordinates are 207 164,
 Name: Player Gennaro Gattuso ITA coordinates are 276 155,
 Name: Player Simone Perrotta ITA coordinates are 336 149,
 Name: Player Gianluca Zambrotta ITA coordinates are 394 148]
[Name: Player Marco Materazzi ITA coordinates are 83 57,
 Name: Player Luca Toni ITA coordinates are 149 59,
 Name: Player Gianluigi Buffon ITA coordinates are 206 62,
 Name: Player Fabio Grosso ITA coordinates are 268 56,
 Name: Player Francesco Totti ITA coordinates are 336 57]
looks like a two column list
Action of type output with variable name output by rule handleUIDStatus
addToOutputQueue(): received wme! WME54 [UpdatePhysicalContext\1]
integrating 1 DiscourseObject(s)
- adding new DO [sportevent#FootballNationalTeam(http://smartweb/kb#WorldCup2006_Team_ITA)]
Current configuration of the Referential Layer:
[sportevent#FootballNationalTeam(http://smartweb/kb#WorldCup2006_Team_ITA)] d0 activation is 0.8
[sportevent#FIFAWorldCup(http://.../individuals/fade#f497)] d0 activation is 0.60
[sportevent#FootballNationalTeam(http://smartweb/kb#WorldCup1990_Team_GER)] d0 activation is 0.40
[sportevent#FIFAWorldCup(http://.../individuals/fade#f451)] d0 activation is 0.20
update physical context finished!

```

User: *“Wer ist der zweite Spieler von rechts oben?”*

```

ResolveMultipleHypotheses(): processing 2 interpretation hypotheses
ResolveMultipleHypotheses(): looping over hypotheses
about to resolve temporal references
hypothesis does not contain any temporal references...
Processing hypothesis: wer ist der dritte Spieler von rechts oben
-> ResolveReferringExpressions(): about to resolve referring expressions
this hypothesis contains 1 referring expressions
about to resolve reference [sportevent#FootballPlayer(http://.../individuals/fade#f523)]
got information about the differentiation criterion 3
first trying hypotheses that this is a reference to the physical context
-> resolveReferenceToTwoColumnList()
has_DCValue is set!
Number of layers 2
number of absolutePos = 5
Startposition is [discourse#AbsolutePosition(http://.../individuals/fade#f510)]
iteration 0
iteration 1
checking id mapping for id http://smartweb/kb#Player_ITA_Buffon_Gianluigi
hypothesis was correct; reference resolution completed
got 1 focus objects
looks like this focus object was embedded in the refProp
replaced the object
Computing score:
nlu score is = 0.6667
npr = 1.0
dpg = 1.0
udr = 1.0
Score of the current hypothesis: 0.916675
about to resolve temporal references
hypothesis does not contain any temporal references...
Processing hypothesis: wer ist der dritte Spieler von rechts oben
-> ResolveReferringExpressions(): about to resolve referring expressions
this hypothesis contains 1 referring expressions
about to resolve reference [sportevent#FieldMatchFootballPlayer(http://.../individuals/fade#f535)]
got information about the differentiation criterion 3
first trying hypotheses that this is a reference to the physical context
-> resolveReferenceToTwoColumnList()
has_DCValue is set!
Number of layers 2
number of absolutePos = 5
Startposition is [discourse#AbsolutePosition(http://.../individuals/fade#f510)]
iteration 0

```



```

iteration 1
checking id mapping for id http://smartweb/kb#Player_ITA_Buffon_Gianluigi
  hypothesis was correct; reference resolution completed
got 1 focus objects
got focused object <object type="sportevent#FieldMatchFootballPlayer" id="http://.../individuals/fade#f535">
  <slot name="linginfo#lingInfo">
    <object type="discourse#RefProp" id="http://.../individuals/fade#f536">
      <slot name="discourse#has_differentiationCriterion">
        <object type="discourse#SpatialReference" id="http://.../individuals/fade#f537">
          <slot name="discourse#relatum">
            <object type="discourse#UpperRight" id="http://.../individuals/fade#f538"/>
          </slot>
          <slot name="discourse#has_DCValue">
            <value type="String">3</value>
          </slot>
        </object>
      </slot>
    </object>
  </slot>
</object>
looks like this focus object was embedded in the refProp
replaced the object
  Computing score:
  nlu score is = 0.6667
  npr = 1.0
  dpq = 1.0
  udr = 1.0
  Score of the current hypothesis: 0.916675
ResolveMultipleHypotheses(): finished; score of winning hypothesis is: 0.916675
Content of wining hypothesis: <object type="discourse#Query" id="http://.../individuals/fade#f521">
  <slot name="emma#id"> <value type="String">1171631878630</value> </slot>
  <slot name="emma#lang"> <value type="String">de</value> </slot>
  <slot name="emma#confidence"> <value type="String">0.6667</value> </slot>
  <slot name="discourse#dialogueAct">
    <object type="discourse#Question" id="http://.../individuals/fade#f527"/>
  </slot>
  <slot name="discourse#focus">
    <object type="discourse#Focus" id="http://.../individuals/fade#f528">
      <slot name="discourse#qeVariable">
        <object type="discourse#QEVariable" id="http://.../individuals/fade#f530">
          <slot name="discourse#varContext">
            <object type="discourse#VarContext" id="http://.../individuals/fade#f531">
              <slot name="discourse#contextObject">
                <object type="sportevent#FootballPlayer"
                  id="http://smartweb/kb#Player_ITA_Buffon_Gianluigi">
                  <slot name="rdf#label">
                    <value type="String">Player Gianluigi Buffon ITA</value>
                  </slot>
                  ...
                <slot name="sportevent#impersonatedBy">
                  <object type="smartsumo#Man" id="http://smartweb/kb#Man_Buffon_Gianluigi">
                    <slot name="rdf#label">
                      <value type="String">Person Gianluigi Buffon</value>
                    </slot>
                    <slot name="smartdolce#HAS-DENOMINATION">
                      <object type="smartdolce#natural-person-denomination"
                        id="http://smartweb/kb#Name_Buffon_Gianluigi">
                        <slot name="smartdolce#LASTNAME">
                          <value type="String">Buffon</value>
                        </slot>
                        <slot name="smartdolce#FIRSTNAME">
                          <value type="String">Gianluigi</value>
                        </slot>
                      </object>
                    </slot>
                  </object>
                </slot>
              </object>
            </slot>
          </object>
        </slot>
      </object>
    </slot>
  </object>
</slot>

```

```

<slot name="discourse#text">
  <value type="String">wer ist der dritte Spieler von rechts oben</value>
</slot>
<slot name="discourse#content">
  <object type="discourse#QEPattern" id="http://.../individuals/fade#f522">
    <slot name="discourse#patternArg">
      <object type="sportevent#FootballPlayer"
        id="http://smartweb/kb#Player_ITA_Buffon__Gianluigi">
        ...
      </object>
    </slot>
  </object>
</slot>
</object>
</slot>
</object>

```

Action of type output with variable name interpretation by rule outputResult(withoutAttentionalStatus)

adding a new user turn to the discourse context

updating global focus

updating local focus: current turn is a user turn

found 1 referring expressions.

found 1 discourseObjects

integrating 1 DiscourseObject(s)

- adding new DO [smartsumo#Man(http://.../individuals/fade#f553)]

Current configuration of the Referential Layer:

[smartsumo#Man(http://.../individuals/fade#f553)] d0 activation is 1.0

[sportevent#FootballNationalTeam(http://smartweb/kb#WorldCup2006_Team_ITA)] d0 activation is 0.60

[sportevent#FIFAWorldCup(http://.../individuals/fade#f497)] d0 activation is 0.40

[sportevent#FootballNationalTeam(http://smartweb/kb#WorldCup1990_Team_GER)] d0 activation is 0.20

System: *"Gianluigi Buffon."*



adding a new system turn to the discourse context

updating global focus

updating local focus: current turn is a system turn

integrating 1 DiscourseObject(s)

- adding new DO [sportevent#FootballPlayer(http://smartweb/kb#Player_ITA_Buffon__Gianluigi)]

Current configuration of the Referential Layer:

[sportevent#FootballPlayer(http://smartweb/kb#Player_ITA_Buffon__Gianluigi)] d0 activation is 1.0

[smartsumo#Man(http://.../individuals/fade#f553)] d0 activation is 0.8

[sportevent#FootballNationalTeam(http://smartweb/kb#WorldCup2006_Team_ITA)] d0 activation is 0.40

[sportevent#FIFAWorldCup(http://.../individuals/fade#f497)] d0 activation is 0.20

Received an update of the configuration of the Physical Environment

looks like a MPEG7-based representation encompassing 0 still regions

about to analyse spatial organization...

Appendix B

FADE's Rule Base for OMDIP

In order to provide an overview of how production rules for FADE look like, we present a selected number of production rules of the OMDIP system. How these rules interact in order to realize the intended functionality of the OMDIP system is described in section 9.4.

```
<rule name="handleStartOfSpeech">
  <comments> Removes a startOfSpeech event from the goal-stack </comments>
  <weighting> 1.0 </weighting>
  <conditions>
    <condition name="goal">
      <object type="emma#Info">
        <slot name="swemma#status">
          <object type="swemma#ASRStatus">
            <slot name="swemma#dispatch">
              <object type="swemma#StartOfRecognition"/>
              <variable name="startOfSpeech"/>
            </slot>
            <slot name="swemma#dispatch"> <empty/> </slot>
          </object>
        </slot>
      </object>
    </condition>
    <condition name="displayRepresentation">
      <object type="discourse#Display"/>
    </condition>
    <condition name="currentPhysicalEnvironment">
      <object type="discourse#PhysicalEnvironment"/>
    </condition>
  </conditions>
  <actions>
    <action type="deleteWME" name="goal"/>
    <action type="addWME" name="startOfSpeech"/>
  </actions>
</rule>

<rule name="initializeDisplayRepresentation">
  <comments> This rule initializes the physical environment of the WM </comments>
  <weighting> 0.5 </weighting>
  <conditions>
    <condition name="goal">
      <object type="discourse#Display">
        <slot name="discourse#has_physicalEnvironment"> <empty/> </slot>
      </object>
    </condition>
    <not-condition name="displayRepresentation">
      <object type="discourse#Display"/>
    </not-condition>
  </conditions>
```

```

    <actions>
      <action type="addWME" name="pE">
        <object type="discourse#PhysicalEnvironment"/>
      </action>
      <action type="pop"/>
    </actions>
  </rule>

  <rule name="updateDisplayRepresentation(1)">
    <comments> Updates an existing physical environment and display representation </comments>
    <weighting> 1.0 </weighting>
    <conditions>
      <condition name="goal" method="restricted">
        <object type="discourse#Display">
          <slot name="discourse#has_physicalEnvironment">
            <object type="discourse#PhysicalEnvironment"/>
            <variable name="pEnv"/>
          </slot>
        </object>
      </condition>
      <condition name="displayRepresentation">
        <object type="discourse#Display"/>
      </condition>
      <condition name="currentPhysicalEnvironment">
        <object type="discourse#PhysicalEnvironment"/>
      </condition>
    </conditions>
    <actions>
      <action type="deleteWME" name="displayRepresentation"/>
      <action type="deleteWME" name="currentPhysicalEnvironment"/>
      <action type="pop"/>
      <action type="addWME" name="pEnv"/>
    </actions>
  </rule>

  <rule name="resolvePointingGesture">
    <comments> Preprocesses a received pointing gesture. </comments>
    <weighting> 1.0 </weighting>
    <conditions>
      <condition name="goal">
        <object type="emma#Interpretation">
          <slot name="emma#applicationInstanceData">
            <object type="discourse#PointingGesture">
              <slot name="discourse#has_referencedObject">
                <empty/>
              </slot>
            </object>
            <variable name="pointingGesture"/>
          </slot>
        </object>
      </condition>
      <condition name="displayRepresentation">
        <object type="discourse#Display"/>
      </condition>
      <function name="de.dfki.dim.plugin.ondip.ResolvePointingGesture" arg1="pointingGesture"
        arg2="displayRepresentation" result="resolved"/>
    </conditions>
    <actions>
      <action type="deleteWME" name="goal"/>
      <action type="push" name="resolved"/>
    </actions>
  </rule>

  <rule name="handlePointingGesture">
    <comments> This rule sends the preprocessed pointing gesture
      discourse context and stores it in the WM. </comments>

```

```

<weighting> 0.5 </weighting>
<conditions>
  <condition name="goal" method="restricted">
    <object type="discourse#PointingGesture">
      <slot name="discourse#has_referencedObject"/>
      <slot name="discourse#has_objectReferenceString"> <empty/> </slot>
    </object>
  </condition>
  <condition name="displayRepresentation">
    <object type="discourse#Display"/>
  </condition>
  <condition name="currentPhysicalEnvironment">
    <object type="discourse#PhysicalEnvironment"/>
  </condition>
</conditions>
<actions>
  <action type="pop"/>
  <action type="output" name="goal" channel="dim"/>
  <action type="updateWME" name="newGoal">
    <object type="SynchronizationPhase">
      <slot name="_timeout">
        <value type="string"> 500 </value>
      </slot>
    </object>
  </action>
  <action type="push" name="newGoal"/>
</actions>
</rule>

<rule name="outputResult">
  <comments> This is the central rule for generating output events that will be sent to the action
    planner. It can only fire when the previous phases have been finished. </comments>
  <weighting> 0.2 </weighting>
  <conditions>
    <goal name="goal" method="unify">
      <object type="EvaluationPhase"/>
    </goal>
    <condition name="interpretation" method="unify">
      <object type="emma#OneOf"/>
    </condition>
    <condition name="displayRepresentation">
      <object type="discourse#Display"/>
    </condition>
    <condition name="currentPhysicalEnvironment">
      <object type="discourse#PhysicalEnvironment"/>
    </condition>
  </conditions>
  <actions>
    <action type="output" name="interpretation" channel="sw"/>
    <action type="output" name="interpretation" channel="dim"/>
    <action type="deleteWME" name="interpretation"/>
    <action type="deleteWME" name="goal"/>
    <action type="addWME" name="cleanUp">
      <object type="CleanUp"/>
    </action>
    <action type="push" name="cleanUp"/>
  </actions>
</rule>

<rule name="cleanUp">
  <comments> Removes all remaing WMEs from the WM. </comments>
  <weighting> 1.0 </weighting>
  <conditions>
    <goal name="goal" method="unify">
      <object type="CleanUp"/>
    </goal>
  </conditions>

```

```

    <condition name="pGesture">
      <object type="discourse#PointingGesture"/>
    </condition>
    <condition name="display">
      <object type="discourse#Display"/>
    </condition>
    <condition name="currentPhysicalEnvironment">
      <object type="discourse#PhysicalEnvironment"/>
    </condition>
  </conditions>
  <actions>
    <action type="deleteWME" name="pGesture"/>
  </actions>
</rule>

<rule name="handleReferences">
  <comments> This rule triggers the contextual interpretation of an hypothesis. </comments>
  <weighting> 0.5 </weighting>
  <conditions>
    <goal name="goal">
      <object type="InterpretationPhase"/>
    </goal>
    <condition name="interpretation">
      <object type="emma#OneOf"/>
    </condition>
  </conditions>
  <actions>
    <function name="de.dfki.dim.plugin.ResolveReferringExpressions"
      arg1="interpretation" result="resolved"/>
    <action type="updateWME" name="interpretation">
      <variable name="resolved"/>
    </action>
    <action type="deleteWME" name="goal"/>
  </actions>
</rule>

<rule name="popInterpretationPhase">
  <comments> Removes the object signaling the interpretation phase
  from the goal-stack when all hypotheses have been interpreted. </comments>
  <weighting> 0.3 </weighting>
  <conditions>
    <goal name="goal">
      <object type="InterpretationPhase"/>
    </goal>
    <not-condition name="interpretation">
      <object type="emma#Interpretation"/>
    </not-condition>
    <not-condition name="pointingGesture">
      <object type="discourse#PointingGesture"/>
    </not-condition>
  </conditions>
  <actions>
    <action type="deleteWME" name="goal"/>
  </actions>
</rule>

```

Bibliography

- Ahrenberg, L., Dahlbäck, N., and Jönsson, A. (1995). Coding Schemes for Studies of Natural Language Dialogue. Technical report, Working Notes from AAAI Spring Symposium, Stanford, CA.
- Ahrenberg, L., Jönsson, A., and Dahlbäck, N. (1991). Discourse Representation and Discourse Management for a Natural Language Dialogue System. Research Report LiTH-IDA-R-91-21, Institutionen för Datavetenskap, Universitetet och Tekniska Högskolan Linköping.
- Alexandersson, J. (2003). *Hybrid Discourse Modelling and Summarization for a Speech-to-Speech Translation System*. PhD thesis, Department of Computer Science, Saarland University.
- Alexandersson, J. and Becker, T. (2001). Overlay as the Basic Operation for Discourse Processing in a Multimodal Dialogue System. In *Proceedings of the 2nd IJCAI Workshop Knowledge and Reasoning in Practical Dialogue System*, pages 8–14, Seattle, WA.
- Alexandersson, J. and Becker, T. (2003). The Formal Foundations Underlying Overlay. In *Proceedings of the 5th International Workshop on Computational Semantics (IWCS-5)*, pages 22–36, Tilburg, The Netherlands.
- Alexandersson, J. and Becker, T. (2006). Efficient Computation of Overlay for Multiple Inheritance Hierarchies in Discourse Modeling. In Bunt, H. and Muskens, R., editors, *Computing Meaning*, volume 3 of *Studies in Linguistics and Philosophy*, pages 423–455. Kluwer, Dordrecht, The Netherlands.
- Alexandersson, J., Becker, T., and Pfeleger, N. (2004). Scoring for Overlay based on Informational Distance. In *Proceedings der 7. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS'04)*, pages 1–4, Vienna, Austria.
- Alexandersson, J., Becker, T., and Pfeleger, N. (2006). Overlay. In Wahlster, W., editor, *SmartKom—Foundations of Multimodal Dialogue Systems*, pages 255–268. Springer, Heidelberg, Germany.
- Alexandersson, J., Buschbeck-Wolf, B., Fujinami, T., Kipp, M., Koch, S., Maier, E., Reithinger, N., Schmitz, B., and Siegel, M. (1998). Dialogue Acts in VERBMOBIL-2 – Second Edition. Verbmobil-Report 226, DFKI Saarbrücken, Universität Stuttgart, Technische Universität Berlin, Universität des Saarlandes.
- Alexandersson, J. and Pfeleger, N. (2006). Discourse Modeling. In Wahlster, W., editor, *SmartKom—Foundations of Multimodal Dialogue Systems*, pages 237–254. Springer, Heidelberg, Germany.

- Allen, J. (1979). *A Plan-Based Approach to Speech Act Recognition*. PhD thesis, Department of Computer Science, University of Toronto.
- Allen, J. (1995). *Natural Language Understanding*. Benjamin/Cummings, Redwood City, CA.
- Allen, J., Blaylock, N., and Ferguson, G. (2002). Problem Solving Model for Collaborative Agents. In Gini, M., Ishida, T., Castelfranchi, C., and Johnson, W. L., editors, *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 774–781, Bologna, Italy. ACM.
- Allen, J., Ferguson, G., and Stent, A. (2001). An Architecture for More Realistic Conversational Systems. In *Proceedings of the 6th International Conference on Intelligent User Interfaces*, pages 1–8, Santa Fe, NM.
- Allwood, J. (1976). *Linguistic Communication as Action and Cooperation*. PhD thesis, Department of Linguistics, Göteborg University.
- Anderson, J. R. (1983). A Spreading Activation Theory of Memory. *Journal of Verbal Learning and Verbal Behavior*, 22:261–295.
- Anderson, J. R. and Lebiere, C. (1998). *The Atomic Components of Thought*. Erlbaum, Mahwah, NJ.
- Argyle, M. (1988). *Bodily Communication*. Methuen, London, UK.
- Ariel, M. (1990). *Accessing Noun-Phrase Antecedents*. Routledge, New York, NY.
- Asher, N. (1993). *Reference to Abstract Objects in Discourse*. Kluwer, Dordrecht, The Netherlands.
- Austin, J. (1962). *How to do Things with Words*. Clarendon Press, Oxford, UK.
- Barwise, J. and Perry, J. (1983). *Situations and Attitudes*. MIT Press, Cambridge, MA.
- Becker, T. (2006). Natural Language Generation with Fully Specified Templates. In Wahlster, W., editor, *SmartKom—Foundations of Multimodal Dialogue Systems*, pages 401–410. Springer, Heidelberg, Germany.
- Becker, T., Blaylock, N., Gerstenberger, C., Korthauer, A., Perera, N., Poller, P., Schehl, J., Steffens, F., Stegmann, R., and Steigner, J. (2006a). D5.3: In-Car Showcase Based on TALK Libraries. Deliverable 5.3, TALK Project.
- Becker, T., Gerstenberger, C., Kruijff-Korbayova, I., Korthauer, A., Pinkal, M., Pitz, M., Poller, P., and Schehl, J. (2006b). Natural and Intuitive Multimodal Dialogue for In-Car Applications: The SAMMIE System. In *Proceedings of the 4th European Conference of Prestigious Applications of Intelligent Systems*, pages 612–616, Riva del Garda, Italy.
- Beckhaus, S., Blom, K., and Haringer, M. (2005). Intuitive, Hands-free Travel Interfaces for Virtual Environments. In *VR2005 Workshop New Directions in 3D User Interfaces*, pages 57–60, Bonn, Germany.

- Bell, A. (1984). Language Style as Audience Design. *Language in Society*, 13:145–204.
- Beringer, N., Oppermann, D., and Steininger, S. (2001). Possible Lexical Indicators for Barge-In/Barge-Before in a Multimodal Man-Maschine-Communication. Technical Report 9, Ludwig-Maximilians-Universität München.
- Bertomeu, N., Uszkoreit, H., Frank, A., Krieger, H.-U., and Jörg, B. (2006). Contextual Phenomena and Thematic Relations in Database QA Dialogues: Results from a Wizard-of-Oz Experiment. In *Proceedings of the HLT-NAACL 2006 Workshop on Interactive Question Answering*, pages 1–8, New York, NY.
- Blaylock, N. J. (2005). *Towards Tractable Agent-based Dialogue*. PhD thesis, Department of Computer Science, University of Rochester.
- Blocher, A. (1999). *Ressourcenadaptierende Raumbeschreibung: Ein beschränkt-optimaler Lokalisationsagent*. PhD thesis, Department of Computer Science, Saarland University.
- Boda, P. P. (2004). A Maximum Entropy Based Approach for Multimodal Integration. In *Proceedings of the 6th International Conference on Multimodal Interfaces (ICMI'04)*, pages 337–338, State College, PA.
- Bolt, R. (1980). Put-that-there: Voice and Gesture at the Graphics Interface. *Computer Graphics*, 14(3):262–270.
- Bouquet, P., Ghidini, C., Giunchiglia, F., and Blanzieri, E. (2003). Theories and Uses of Context in Knowledge Representation and Reasoning. *Journal of Pragmatics*, 35(3):455–484.
- Boves, L., Neumann, A., Vuurpijl, L., ten Bosch, L., Rossignol, S., Engel, R., and Pfleger, N. (2004). Multimodal Interaction in Architectural Design Applications. In *Proceedings of 8th ERCIM Workshop on User Interfaces for All*, pages 384–390, Vienna, Austria.
- Boye, J., Wiren, M., and Gustafson, J. (2004). Contextual Reasoning in Multimodal Dialogue Systems: Two Case Studies. In *Proceedings of the 8th Workshop on the Semantics and Pragmatics of Dialogue (Catalogue 2004)*, pages 4–11, Barcelona, Spain.
- Bratman, M. E. (1987). *Intention, Plans, and Practical Reason*. CSLI Publications, Stanford, CA.
- Bregler, C., Hild, H., Manke, S., and Waibel, A. (1993). Improving Connected Letter Recognition by Lipreading. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (IEEE-ICASSP)*, pages 557–560, Minneapolis, MN.
- Buitelaar, P., Declerck, T., Frank, A., Racioppa, S., Kiesel, M., Sintek, M., Romanelli, M., Engel, R., Sonntag, D., Loos, B., Micelli, V., Porzel, R., and Cimiano, P. (2006). LingInfo: A Model for the Integration of Linguistic Information in Ontologies. In *Proceedings of the Workshop on Interfacing Ontologies and Lexical Resources for Semantic Web Technologies (OntoLex 2006)*, pages 28–32, Genova, Italy.
- Bunescu, R. (2003). Associative Anaphora Resolution: A Web-Based Approach. In *Proceedings of the EACL-2003 Workshop on the Computational Treatment of Anaphora*, pages 47–52, Budapest, Hungary.

- Bunt, H. (2000). Dialogue Pragmatics and Context Specification. In Bunt, H. and Black, W., editors, *Abduction, Belief and Context in Dialogue*, volume 1 of *Natural Language Processing*, pages 81–150. John Benjamins, Amsterdam, The Netherlands.
- Bunt, H., Kipp, M., Maybury, M., and Wahlster, W. (2005). Fusion and Coordination for Multimodal Interactive Information Presentation. In Stock, O. and Zancanaro, M., editors, *Multimodal Intelligent Information Presentation. Series Text, Speech and Language Technology*, volume 27, pages 325–340. Kluwer, Dordrecht, The Netherlands.
- Bunt, H. C. (1994). Context and Dialogue Control. *Think*, 3:19–31.
- Bußmann, H. (2002). *Lexikon der Sprachwissenschaft*. Kröner, Stuttgart, Germany, 3., updated and extended edition.
- Byron, D. K., Mampilly, T., Sharma, V., and Xu, T. (2005). Utilizing Visual Attention for Cross-Modal Coreference Interpretation. In *Lecture Notes in Computer Science: Proceedings of Context-05*, pages 83–96, Heidelberg, Germany. Springer.
- Cappella, J. N. (1985). Controlling the Floor in Conversation. In Siegman, A. W. and Feldstein, S., editors, *Multichannel Integrations of Nonverbal Behavior*, pages 69–103. Erlbaum, Hillsdale, NJ.
- Carbonell, J. G. (1983). Discourse Pragmatics and Ellipsis Resolution in Task-Oriented Natural Language Interfaces. In *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics (ACL'83)*, pages 164–169, Department of Computer Science University of Massachusetts.
- Carlett, J., Anderson, A. H., and Garrod, S. (2002). Seeing Eye to Eye: An Account of Grounding and Understanding in Work Groups. In *Cognitive Studies: Bulletin of the Japanese Cognitive Science Society*, volume 9(1), pages 1–20.
- Carletta, J., Isard, A., Isard, S., Kowtko, J. C., Doherty-Sneddon, G., and Anderson, A. H. (1997). The Reliability of a Dialogue Structure Coding Scheme. *Computational Linguistics*, 23(1):13–31.
- Carpenter, B. (1992). *The Logic of Typed Feature Structures*. Cambridge University Press, Cambridge, England.
- Cassell, J. (2000). More than Just Another Pretty Face: Embodied Conversational Interface Agents. *Communications of the ACM*, 43(4):70–78.
- Cassell, J., Bickmore, T., Campbell, L., Vilhjalmsson, H., and Yan, H. (2001). Human Conversation as a System Framework: Designing Embodied Conversational Agents. In Cassell, J., Sullivan, J., and Prevost, S., editors, *Embodied Conversational Agents*, pages 29–63. MIT Press, Cambridge, MA.
- Cassell, J., Torres, O., and Prevost, S. (1999). Turn Taking vs. Discourse Structure: How Best to Model Multimodal Conversation. In Wilks, Y., editor, *Machine Conversations*, pages 143–154. Kluwer, The Hague.

- Chai, J. Y. and Jin, R. (2004). Discourse Structure for Context Question Answering. In *HLT-NAACL 2004 Workshop on Pragmatics in Question Answering*, pages 23–30, Boston, MA. ACL.
- Chibelushi, C. C. and Bourel, F. (2006). Facial Expression Recognition: A Brief Tutorial Overview. In *CVonline: On-Line Compendium of Computer Vision [Online]*. Available: <http://homepages.inf.ed.ac.uk/rbf/CVonline/>. Last access [15.10.2006].
- Clark, H. and Carlson, T. B. (1982). Hearers and Speech Acts. *Language*, 58:332–373.
- Clark, H. and Schaefer, E. (1989). Contributing to Discourse. *Cognitive Science*, (13):259–294.
- Clark, H. H. (1977). Bridging. In Johnson-Laird, P. N. and Wason, P. C., editors, *Thinking: Readings in Cognitive Science*, pages 169–174. Cambridge University Press, Cambridge, England.
- Clark, H. H. (1996). *Using Language*. The Press Syndicate of the University of Cambridge, Cambridge, UK.
- Clark, H. H. and Brennan, S. E. (1991). Grounding in Communication. In Resnick, L. B., Levine, J., and Teasley, S. D., editors, *Perspectives on Socially Shared Cognition*, pages 127–149. American Psychological Association.
- Clark, H. H. and Haviland, S. (1974). Psychological Processes as Linguistic Explanation. In Carrol, J. and Freedle, R., editors, *Explaining Linguistic Phenomena*, pages 91–124. Hemisphere Publishing Corp., Washington.
- Cohen, P. R., Johnston, M., McGee, D., Oviatt, S. L., Pittman, J. A., Smith, I., Chen, L., and Clow, J. (1997). QuickSet: Multimodal Interaction for Distributed Applications. In *Proceedings of ACM Multimedia 1997*, pages 31–40, Seattle, WA.
- Collins, A. and Loftus, E. (1975). A Spreading-Activation Theory of Semantic Processing. *Psychological Review*, 82(6):407–428.
- da Figueira Filho, C. S. and Ramalho, G. (2000). JEOPS—The Java Embedded Object Production System. In *Proceedings of the International Joint Conference, 7th Ibero-American Conference on AI (IBERAMIA-SBIA '00)*, pages 53–62, London, UK. Springer-Verlag.
- Dale, R. (1992). *Generating Referring Expressions: Constructing Descriptions in a Domain of Objects and Processes*. MIT Press, Cambridge, MA.
- de Ruiter, J. P. (2003). Research on Modality Effects on Performance Quality and Efficiency. Technical report, Max Planck Institute for Psycholinguistics, Nijmegen, The Netherlands.
- den Os, E. and Boves, L. (2003). Towards Ambient Intelligence: Multimodal Computers that Understand Our Intentions. In *eChallenges e-2003*, pages 22–24, Vienna, Austria.
- Doherty, P., Granlund, G., Kuchcinski, K., Sandewall, E., Nordberg, K., Skarman, E., and Wiklund, J. (2000). The WITAS Unmanned Aerial Vehicle Project. In Horn, W., editor, *ECAI 2000. Proceedings of the 14th European Conference on Artificial Intelligence*, pages 747–755, Berlin, Germany.

- Dunbar, R., Duncan, N., and Nettle, D. (1995). Size and Structure of Freely Forming Conversational Groups. *Human Nature*, 6(1):67–78.
- Duncan, S. (1972). Some Signals and Rules for Taking Speaking Turns in Conversations. *Journal of Personality and Social Psychology*, 23(2):283–292.
- Eckert, M. (1998). The German Topic Position and Null Anaphora. In *Proceedings of the Discourse, Anaphora and Reference Colloquium (DAARC2)*. University of Lancaster Technical Report.
- Edlund, J., Skantze, G., and R., C. (2004). Higgins—a Spoken Dialogue System for Investigating Error Handling Techniques. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP'04)*, pages 229–231, Jeju, Korea.
- Efron, D. (1941). *Gesture and Environment*. King's Crown Press, New York, NY.
- Ekman, P. (1965). Communication Through Nonverbal Behavior: A Source of Information About an Interpersonal Relationship. In Tomkins, S. S. and Izard, C. E., editors, *Affect, Cognition and Personality*. Springer, New York, NY.
- Eliasson, K. (2005). Integrating a Discourse Model with a Learning Case-Based Reasoning System. In *Proceedings of the 9th Workshop on the Semantics and Pragmatics of Dialogue (SEMDIAL)*, pages 29–36, Nancy, France.
- Ellis, A. and Beattie, G. (1986). *The Psychology of Language and Communication*. Weidenfeld, London, UK.
- Engel, R. (2002). SPIN: Language Understanding for Spoken Dialogue Systems Using a Production System Approach. In *Proceedings of 7th International Conference on Spoken Language Processing (ICSLP'02)*, pages 2717–2720, Denver, CO.
- Engel, R. (2005). Robust and Efficient Semantic Parsing of Free Word Order Languages in Spoken Dialogue Systems. In *Proceedings of the 9th European Conference on Speech Communication and Technology (Interspeech/Eurospeech)*, pages 3461–3464, Lisbon, Portugal.
- Engel, R. (2006). Natural Language Understanding. In Wahlster, W., editor, *SmartKom—Foundations of Multimodal Dialogue Systems*, pages 195–209. Springer, Heidelberg, Germany.
- Engel, R. and Pflieger, N. (2006). Multimodal Fusion. In Wahlster, W., editor, *SmartKom—Foundations of Multimodal Dialogue Systems*, pages 223–236. Springer, Heidelberg, Germany.
- Eysenck, M. W. and Keane, M. T. (2000). *Cognitive Psychology. A Student's Handbook*. Psychology Press Ltd, Philadelphia, PA, 4th edition.
- Fan, J., Barker, K., and Porter, B. W. (2005). Indirect Anaphora Resolution as Semantic Path Search. In *Proceedings of the 3rd International Conference on Knowledge Capture (K-CAP)*, pages 153–160, New York, NY. ACM.
- Fensel, D., Horrocks, I., van Harmelen, F., McGuinness, D. L., and Patel-Schneider, P. F. (2001). OIL: An Ontology Infrastructure for the Semantic Web. *IEEE Intelligent Systems*, 16(2):38–45.

- Filisko, E. and Seneff, S. (2003). A Context Resolution Server for the Galaxy Conversational Systems. In *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech)*, pages 197–200, Geneva, Switzerland.
- Fillmore, C. (1971). Towards a Theory of Deixis. In *The PCCLLU Papers*, pages 219–241. Department of Linguistics, University of Hawaii.
- Fischer, K. (2000). Discourse Particles, Turn-taking, and the Semantics-Pragmatics Interface. *Revue de Semantique et Pragmatique*, 8(11):111–137.
- Fleischman, M. and Roy, D. (2005). Intentional Context in Situated Language Learning. In *Ninth Conference on Computational Natural Language Learning*, pages 104–111, Ann Arbor, MI.
- Fuhr, T., Socher, G., Scheering, C., and Sagerer, G. (1995). A Three-Dimensional Spatial Model for the Interpretation of Image Data. In *Proceedings of the Workshop on Representation and Processing of Spatial Expressions (IJCAI'95)*, pages 103–118, Montreal, Canada.
- Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., and Schneider, L. (2002). Sweetening Ontologies with DOLCE. In *13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02)*, volume 2473 of *Lecture Notes in Computer Science*, pages 166–182, Sigüenza, Spain.
- Gapp, K.-P. (1996). *Ein Objektlokalisierungssystem zur sprachlichen Raumbeschreibung in dreidimensionalen Umgebungen: Formalisierung, Implementierung und empirische Validierung*. PhD thesis, Department of Computer Science, Saarland University.
- Gebhard, P. (2005). ALMA—A Layered Model of Affect. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 29–36, Utrecht, The Netherlands.
- Gehrmann, T. (2007). Der Einsatz von Spreading Activation zur Bestimmung von Informationsclustern in Dialogsystemen. Master's thesis, Department of Computer Science, Saarland University.
- Giunchiglia, F. (1993). Contextual Reasoning. *Epistemologia—Special Issue on "I Linguaggi e le Macchine" XVI*, pages 345–364.
- Göbel, S., Schneider, O., Iurgel, I., Feix, A., Knöpfle, C., and Rettig, A. (2006). VirtualHuman: Storytelling and Computer Graphics for a Virtual Human Platform. In *Proceedings of the 2nd International Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE 2004)*, pages 79–88, Darmstadt, Germany. Springer.
- Goodwin, C. (1981). *Conversational Organization: Interaction between Hearers and Speakers*. Academic Press, New York, NY.
- Görz, G., Rollinger, C.-R., and Schneeberger, J., editors (2003). *Handbuch der künstlichen Intelligenz*. Oldenbourg, München, Germany.
- Grice, H. (1975). Logic and Conversation. In Cole, P. and Morgan, J., editors, *Syntax and Semantics*, volume 3, pages 42–58. Academic Press, New York, NY.

- Grishman, R. and Kittredge, R. I. (1986). *Analysing Language in Restricted Domains*. Erlbaum, Hillsdale, NJ.
- Grosz, B., Joshi, A. K., and Weinstein, S. (1995). Centering: A Framework for Modelling the Local Coherence of Discourse. *Computational Linguistics*, 21(2):203–225.
- Grosz, B. J. (1977). The Representation and Use of Focus in a System for Understanding Dialogs. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'77)*, pages 67–76, Cambridge, MA.
- Grosz, B. J. and Sidner, C. L. (1986). Attention, Intentions, and the Structure of Discourse. *Computational Linguistics*, 12(3):175–204.
- Grosz, B. J. and Sidner, C. L. (1990). Plans for Discourse. In Cohen, P. R., Morgan, J. L., and Pollack, M. E., editors, *Intentions and Communication*, pages 417–444. MIT Press, Cambridge, MA.
- Gruber, T. R. (1993). A Translation Approach to Portable Ontologies. *Knowledge Acquisition*, 5(2):199–220.
- Guarino, N. and Welty, C. A. (2000). A Formal Ontology of Properties. In *12th International Conference on Knowledge Acquisition, Modeling and Management*, pages 97–112, Juan-les-Pins, France.
- Gundel, J., Hedberg, N., and Zacharski, R. (1993). Cognitive Status and the Form of Referring Expressions in Discourse. *Language*, 69:274–307.
- Gupta, A. (2003). An Adaptive Approach to Collecting Multimodal Input. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL'03)*, pages 31–36, Sapporo, Japan. ACL.
- Gurevych, I., Merten, S., and Porzel, R. (2003a). Automatic Creation of Interface Specifications from Ontologies. In *SEALTS '03: Proceedings of the HLT-NAACL 2003 Workshop on Software engineering and Architecture of Language Technology Systems*, pages 59–66, Morristown, NJ. ACM.
- Gurevych, I., Porzel, R., and Malaka, R. (2006). Modeling Domain Knowledge: Know-How and Know-What. In Wahlster, W., editor, *SmartKom—Foundations of Multimodal Dialogue Systems*, pages 71–84. Springer, Heidelberg, Germany.
- Gurevych, I., Porzel, R., Slinko, E., Pflieger, N., Alexandersson, J., and Merten, S. (2003b). Less is More: Using a Single Knowledge Representation in Dialogue Systems. In *Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning*, pages 14–21, Morristown, NJ. ACL.
- Gustafson, J., Bell, L., Beskow, J., Boye, J., Carlson, R., Edlund, J., Granström, B., House, D., and Wiren, M. (2000). AdApt—a Multimodal Conversational Dialogue System in an Apartment Domain. In *Proceedings of the 6th International Conference of Spoken Language Processing (ICSLP '00)*, volume 2, pages 134–137, Beijing, China.
- Gustafson, J., Bell, L., Boye, J., Lindström, A., and Wiren, M. (2004). The NICE Fairy-tale Game System. In *Proceedings of the 5th SIGDial Workshop*, Boston, MA.

- Hacker, C., Batliner, A., and Nöth, E. (2006). Are You Looking at Me, are You Talking with Me – Multimodal Classification of the Focus of Attention. In Sojka, P., Kopeček, I., and Pala, K., editors, *Text, Speech and Dialogue. 9th International Conference (TSD 2006)*, pages 581–588, Heidelberg, Germany. Springer.
- Hahn, U. and Strube, M. (1997). Centering-in-the-Large: Computing Referential Discourse Segments. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and the 8th Conference of the European Chapter of the Association for Computational Linguistics (ACL-EACL'97)*, pages 104–111, Madrid, Spain. ACL.
- Halliday, M. and Hasan, R. (1985). *Language, Context and Text: Aspects of Language in a Social-semiotic Perspective*. Deakin University Press, Victoria, Australia.
- Hardt, D. (1999). Dynamic Interpretation of Verb Phrase Ellipsis. *Linguistics and Philosophy*, 22(2):187–221.
- Heckmann, D. (2006). *Ubiquitous User Modeling*. Akademische Verlagsgesellschaft, Berlin, Germany.
- Herfet, T., Kirste, T., and Schnaider, M. (2001). EMBASSI: Multimodal Assistance for Universal Access to Infotainment and Service Infrastructures. In *Proceedings of the 2001 EC/NSF Workshop on Universal Accessibility of Ubiquitous Computing: Providing for the Elderly*, pages 41–50, Alcaccer do Sal, Portugal.
- Herzog, G. and Ndiaye, A. (2006). Building Multimodal Dialogue Applications: System Integration in SmartKom. In Wahlster, W., editor, *SmartKom—Foundations of Multimodal Dialogue Systems*, pages 439–452. Springer, Heidelberg, Germany.
- Hill, R. W., Gratch, J., Marsella, S., Rickel, J., Swartout, W., and Traum, D. (2003). Virtual Humans in the Mission Rehearsal Exercise System. *KI Special Issue on Embodied Conversational Agents*, 4(3):5–10.
- Hobbs, J. R. (1979). Coherence and Coreference. *Cognitive Science*, 3(1):67–90.
- Holzapfel, H., Nickel, K., and Stiefelhagen, R. (2004). Implementation and Evaluation of a Constraint Based Multimodal Fusion System for Speech and 3D Pointing Gestures. In *Proceedings of the 6th International Conference on Multimodal Interfaces (ICMI'04)*, pages 175–182, State College, PA.
- Homas, G. C. (1951). *The Human Group*. Routledge & Kegan Paul, London.
- Horrocks, I. (1998). Using an Expressive Description Logic: FaCT or Fiction? In Cohn, A. G., Schubert, L., and Shapiro, S. C., editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the 6th International Conference (KR'98)*, pages 636–647, San Francisco, CA. Morgan Kaufmann Publishers.
- Hulstijn, J. (2003). Roles in Dialogue. In *Proceedings the 7th Workshop on the Semantics and Pragmatics of Dialogue (Diabrock)*, pages 43–50, Wallerfangen, Germany.
- Ishii, H. and Ullmer, B. (1997). Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. In *Proceedings of the Conference of Human Factors in Computer Systems (CHI)*, pages 234–241, New York, NY. ACM.

- Jackendoff, R. (2003). *Foundations of Language*. Oxford University Press, Oxford, UK.
- Jameson, A. and Wahlster, W. (1982). User Modelling in Anaphora Generation: Ellipsis and Definite Description. In *Proceedings of the First European Conference on Artificial Intelligence (ECAI)*, pages 133–138, Orsay, France.
- Johnston, M., Bangalore, S., Vasireddy, G., Stent, A., Ehlen, P., Walker, M., Whittaker, S., and Maloor, P. (2001). MATCH: An Architecture for Multimodal Dialogue Systems. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL'02)*, pages 376–383, Morristown, NJ. ACL.
- Johnston, M., Cohen, P. R., McGee, D., Oviatt, S. L., Pittman, J. A., and Smith, I. (1997). Unification based Multimodal Integration. In *Proceedings of the 35th Annual Conference of the Association for Computational Linguistics (ACL'97)*, pages 281–288, Madrid, Spain.
- Jovanovic, N. and op den Akker, R. (2004). Towards Automatic Addressee Identification in Multi-party Dialogues. In *Proceedings 5th SIGDial Workshop on Discourse and Dialogue*, pages 63–71, Boston, MA.
- Jovanovic, N., op den Akker, R., and Nijholt, A. (2006). Addressee Identification in Face-to-Face Meetings. In *11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 169–176, Trento, Italy.
- Jurafsky, D. and Martin, J. H. (2000). *Speech and Language Processing*. Prentice Hall, New Jersey.
- Kamp, H. and Reyle, U. (1993). *From Discourse to Logic*. Kluwer, Dordrecht, Boston, London.
- Kehler, A. (1993). The Effect of Establishing Coherence in Ellipsis and Anaphora Resolution. In *Proceedings of the 31st Annual Conference of the Association for Computational Linguistics (ACL'93)*, pages 62–69, Columbus.
- Kehler, A. (2000). Cognitive Status and Form of Reference in Multimodal Human-Computer Interaction. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-2000)*, pages 685–690, Austin, Texas.
- Kehler, A. (2002). *Coherence, Reference, and the Theory of Grammar*. CSLI lecture notes. CSLI Publishing, Stanford, CA.
- Kempe, B. (2004). *PATE—a Production Rule System based on Activation and Typed Feature Structure Elements*. Bachelor thesis, Department of Computer Science, Saarland University.
- Kempe, B. (2005). Generation of Verbal and Nonverbal Utterances for Embodied Virtual Characters. Master's thesis, Department of Computer Science, Saarland University.
- Kempe, B., Pfleger, N., and Löckelt, M. (2005). Generating Verbal and Nonverbal Utterances for Virtual Characters. In *Proceedings of the International Conference on Virtual Storytelling 2005*, pages 73–78, Strasbourg, France.
- Kendon, A. (1987). On Gesture: Its Complementary Relationship with Speech. In Siegman, A. W. and Feldstein, S., editors, *Nonverbal Behavior and Communication*, pages 65–97. Erlbaum.

- Kendon, A. (1990). Conducting Interaction. In *Some Functions of Gaze Direction in Two-person Conversation*, pages 51–90. The Press Syndicate of the University of Cambridge, Cambridge, UK. First published in *Acta Psychologica*, 26 (1967), 22–63.
- Kendon, A. (2004). *Gesture: Visible Action as Utterance*. The Press Syndicate of the University of Cambridge, Cambridge, UK.
- Kipp, M. (2004). *Gesture Generation by Imitation—From Human Behavior to Computer Character Animation*. Dissertation.com, Boca Raton, FL.
- Knapp, M. L. and Hall, J. A. (2002). *Nonverbal Communication in Human Interaction*. Wadsworth/Thomson Publishing, South Melbourne, Victoria.
- Kopp, S., Jung, B., Lessmann, N., and Wachsmuth, I. (2003). Max—A Multimodal Assistant in Virtual Reality Construction. *KI-Künstliche Intelligenz*, 4:11–17.
- Krauss, R. M., Chen, Y., and Chawla, P. (1996). Nonverbal Behavior and Nonverbal Communication: What do Conversational Hand Gestures Tell Us? In Zanna, M., editor, *Advances in Experimental Social Psychology*, pages 389–450. Academic Press, San Diego, CA.
- Krieger, H.-U. (1995). *TDL—A Type Description Language for Constraint-Based Grammars. Foundation, Implementation, and Application*. PhD thesis, Department of Computer Science, Saarland University.
- Laird, J., Newell, A., and Rosenbloom, P. (1987). Soar: An Architecture for General Intelligence. *Artificial Intelligence*, 33(11):1–64.
- Landragin, F., Bellalem, N., and Romary, L. (2002). Referring to Objects with Spoken and Haptic Modalities. In *Proceedings of the 4th International Conference on Multimodal Interfaces (ICMI'02)*, pages 99–104, Los Alamitos, CA. IEEE CS Press.
- Lappin, S. (1996). The Interpretation of Ellipsis. In Lappin, S., editor, *Handbook of Contemporary Semantic Theory*, pages 145–175. Blackwell Publishers, Oxford, UK.
- Lerner, G. (2003). Selecting Next Speaker: The Context-sensitive Operation of a Context-free Organization. *Language in Society*, 32(2):177–201.
- Lerner, G. H. (1993). Collectivities in Action: Establishing the Relevance of Conjoined Participation in Conversation. *Text*, 13(2):213–245.
- Levinson, S. C. (1983). *Pragmatics*. Press Syndicate of the University of Cambridge, Cambridge, UK.
- Levinson, S. C. (2003). *Space in Language and Cognition*. Press Syndicate of the University of Cambridge, Cambridge, UK.
- Lochbaum, K. E. (1998). A Collaborative Planning Model of Intentional Structure. *Computational Linguistics*, 24(4):525–572.
- Löckelt, M. (2005). Action Planning for Virtual Human Performances. In *Proceedings of the 3rd International Conference on Virtual Storytelling*, pages 53–62, Berlin, Germany. Springer.

- Löckelt, M. (2007). *A Flexible and Reusable Framework for Dialogue and Action Management in Multi-Party Discourse*. PhD thesis, Department of Computer Science, Saarland University. (to appear).
- Löckelt, M., Becker, T., Pflieger, N., and Alexandersson, J. (2002). Making Sense of Partial. In *Proceedings of the 6th Workshop on the Semantics and Pragmatics of Dialogue (Edilog)*, pages 101–107, Edinburgh, UK.
- Löckelt, M., Pecourt, E., and Pflieger, N. (2005). Balancing Narrative Control and Autonomy for Virtual Characters in a Game Scenario. In *Proceedings of the Conference on INtelligent TEchnologies for interactive enterTAINment (INTETAIN 2005)*, pages 248–252, Madonna di Campiglio, Italy.
- Löckelt, M. and Pflieger, N. (2005). Multi-Party Interaction With Self-Contained Virtual Characters. In *Proceedings of the 9th Workshop on the Semantics and Pragmatics of Dialogue (Dialor)*, pages 139–142, Nancy, France.
- Löckelt, M. and Pflieger, N. (2006). Augmenting Virtual Characters for more Natural Interaction. In *Proceedings of the 3rd International Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE 2006)*, pages 231–240, Berlin/Heidelberg, Germany. Springer.
- Löckelt, M., Pflieger, N., and Reithinger, N. (2007). Multiparty Concersation for Mixed Reality. *International Journal of Virtual Reality*, 6(4):31–42.
- Luperfoy, S. (1991). *Discourse Pegs: A Computational Analysis of Context-Dependent Referring Expressions*. PhD thesis, University of Texas at Austin.
- Luperfoy, S. (1992). The Representation of Multimodal User Interface Dialogues Using Discourse Pegs. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics (ACL'92)*, pages 22–31, Newark, NJ.
- Maass, W. (1996). *Von visuellen Daten zu inkrementellen Wegbeschreibungen in dreidimensionalen Umgebungen: Das Modell eines kognitiven Agenten*. Dissertationen zur Künstlichen Intelligenz (DISKI). Akademische Verlagsgesellschaft, Berlin, Germany.
- Mann, W. C. (2002). Dialogue Macrogame Theory. In *Proceedings of the 6th Workshop on Semantics and Pragmatics of Dialogue (Edilog)*, pages 109–116, Edinburgh, UK.
- Mann, W. C. and Thompson, S. A. (1987). Rhetorical Structure Theory: A Theory of Text Organization. In Polanyi, L., editor, *The Structure of Discourse*. Ablex Publishing Corporation.
- Marcel, S. (2002). Gestures For Multi-Modal Interfaces: A Review. Technical Report IDIAP RR 02-34, IDIAP Research Institute.
- Martin, D. L., Cheyer, A. J., and Moran, D. B. (1999). The Open Agent Architecture: A Framework for Building Distributed Software Systems. *Applied Artificial Intelligence*, 13(1-2):91–128.
- Maybury, M. T. and Wahlster, W., editors (1998). *Readings in Intelligent User Interfaces*. Morgan Kaufmann, San Francisco, CA.

- McGee, D. R. and Cohen, P. R. (2001). Creating Tangible Interfaces by Augmenting Physical Objects with Multimodal Language. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI 2001)*, pages 113–119, Santa Fe, NM. ACM.
- McGee, D. R., Cohen, P. R., Wesson, R. M., and Horman, S. (2002). Comparing Paper and Tangible Multimodal Tools. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI'02)*, pages 407–417, Minneapolis, MI. ACM.
- McNeill, D. (1992). *Hand and Mind: What Gestures Reveal About Thought*. The University of Chicago Press, Chicago, IL.
- Mitkov, R. (1998). Robust Pronoun Resolution with Limited Knowledge. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING'98/ACL'98)*, pages 869–975, Montreal, Canada.
- Müller, C. (2006). *Zweistufige kontextsensitive Sprecherklassifikation am Beispiel von Alter und Geschlecht*. Dissertationen zur Künstlichen Intelligenz (DISKI). Akademische Verlags Gesellschaft, Berlin, Germany.
- Nakano, Y. I., Reinstein, G., Stocky, T., and Cassell, J. (2003). Towards a Model of Face-to-Face Grounding. In *Proceedings of Association for Computational Linguistics (ACL'03)*, pages 553–561, Sapporo, Japan.
- Ndiaye, A., Gebhard, P., Kipp, M., Klesen, M., Schneider, M., and Wahlster, W. (2005). Ambient Intelligence in Edutainment: Tangible Interaction with Life-Like Exhibit Guides. In *Proceedings of the Conference on INtelligent TEchnologies for interactive enterTAINment (INTETAIN 2005)*, pages 104–113, Madonna di Campiglio, Italy.
- Niles, I. and Pease, A. (2001). Towards a Standard Upper Ontology. In Welty, C. and Smith, B., editors, *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*, pages 2–9, Ogunquit, ME.
- Oberle, D., Ankolekar, A., Hitzler, P., Cimiano, P., Sintek, M., Kiesel, M., Mougouie, B., Vembu, S., Baumann, S., Romanelli, M., Buitelaar, P., Engel, R., Sonntag, D., Reithinger, N., Loos, B., Porzel, R., Zorn, H.-P., Micelli, V., Schmidt, C., Weiten, M., Burkhardt, F., and Zhou, J. (2006). DOLCE ergo SUMO: On Foundational and Domain Models in SWIntO (SmartWeb Integrated Ontology). Technical report, AIFB, University of Karlsruhe.
- Oviatt, S. (1999a). Mutual Disambiguation of Recognition Errors in a Multimodal Architecture. In *Proceedings of the Conference of Human Factors in Computing Systems (CHI)*, pages 576–583, New York, N.Y. ACM.
- Oviatt, S. (1999b). Ten Myths of Multimodal Interaction. *Communications of the ACM*, 42(11):74–81.
- Oviatt, S., Coulston, R., and Lunsford, R. (2004). When Do We Interact Multimodally? Cognitive Load and Multimodal Communication Patterns. In *Proceedings of the 6th International Conference on Multimodal Interfaces (ICMI'04)*, pages 129–136, State College, PA.

- Oviatt, S., Coulston, R., Tomko, S., Xiao, B., Lunsford, R., Wesson, M., and Carmichael, L. (2003). Toward a Theory of Organized Multimodal Integration Patterns During Human-Computer Interaction. In *Proceedings of the 5th International Conference on Multimodal interfaces (ICMI'03)*, pages 44–51. ACM.
- Oviatt, S. L., DeAngeli, A., and Kuhn, K. (1997). Integration and Synchronization of Input Modes during Multimodal Human-Computer Interaction. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI'97)*, pages 415–422, New York, N.Y. ACM.
- Oxford English Dictionary (2006). Gesture. (Online; last accessed 05.01.2007, <http://www.askoxford.com/>).
- Pavlovic, V. (1998). Multimodal Tracking and Classification of Audio-Visual Features. In *Proceedings of the International Conference on Image Processing (ICIP'98)*, pages 343–347, Chicago, IL.
- Pfalzgraf, A. (2006). Ein robustes System zur automatischen Komposition semantischer Web Services in SmartWeb. Master's thesis, Department of Computer Science, Saarland University.
- Pfleger, N. (2002). Discourse Processing for Multimodal Dialogues and its Application in Smartkom. Diploma thesis, Department of Computational Linguistics, Saarland University.
- Pfleger, N. (2004). Context Based Multimodal Fusion. In *Proceedings of the 6th International Conference on Multimodal Interfaces (ICMI'04)*, pages 265–272, State College, PA.
- Pfleger, N. (2005). FADE - An Integrated Approach to Multimodal Fusion and Discourse Processing. In *Proceedings of the Doctoral Spotlight Session of the International Conference on Multimodal Interfaces (ICMI'05)*, pages 17–21, Trento, Italy.
- Pfleger, N. and Alexandersson, J. (2004). Modeling Non-Verbal Behavior in Multimodal Conversational Systems. *it - Information Technology Special Issue on Conversational User Interfaces*, 46(6):341–345.
- Pfleger, N. and Alexandersson, J. (2006). Towards Resolving Referring Expressions by Implicitly Activated Referents in Practical Dialogue Systems. In *Proceedings of the 10th Workshop on the Semantics and Pragmatics of Dialogue (Brandial)*, pages 2–9, Postdam, Germany.
- Pfleger, N., Alexandersson, J., and Becker, T. (2002). Scoring Functions for Overlay and their Application in Discourse Processing. In *Proceedings der 6. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS'02)*, pages 139–146, Saarbrücken, Germany.
- Pfleger, N., Alexandersson, J., and Becker, T. (2003a). A Robust and Generic Discourse Model for Multimodal Dialogue. In *Proceedings of the Workshop on Knowledge and Reasoning in Practical Dialogue Systems (IJCAI'03)*, pages 64–70, Acapulco, Mexico.
- Pfleger, N., Engel, R., and Alexandersson, J. (2003b). Robust Multimodal Discourse Processing. In *Proceedings of the 7th Workshop on the Semantics and Pragmatics of Dialogue (Diabruck)*, pages 107–114, Wallerfangen, Germany.

- Pfleger, N. and Löckelt, M. (2005). Synchronizing Dialogue Contributions of Human Users and Virtual Characters in a Virtual Reality Environment. In *Proceedings of 9th European Conference on Speech Communication and Technology (Interspeech/Eurospeech)*, pages 2773–2776, Lisbon, Portugal.
- Pfleger, N. and Löckelt, M. (2006). A Comprehensive Context Model for Multi-party Interactions with Virtual Characters. In *Proceedings of the 6th International Conference on Intelligent Virtual Agents (IVA'06)*, pages 157–168, Marina del Rey, CA.
- Pfleger, N. and Schehl, J. (2006). Development of Advanced Dialog Systems with PATE. In *Proceedings of the International Conference on Spoken Language Processing (Interspeech/ICSLP)*, pages 1778–1781, Pittsburgh, PA.
- Pinkal, M. (1977). *Kontext und Bedeutung. Ein probabilistisch erweiterter pragmatischer Beschreibungsansatz*. Gunter Narr, Tübingen, Germany.
- Poesio, M. and Traum, D. (1995). A Multi-Purpose Model of Conversational Context. In *Proceedings of the Workshop on Context in Natural Language Processing (IJCAI'95)*, pages 103–111, Montreal, Canada.
- Porzel, R., Zorn, H.-P., Loos, B., and Malaka, R. (2006). Towards a Separation of Pragmatic Knowledge and Contextual Information. In *Proceedings of ECAI 06 Workshop on Contexts and Ontologies*, pages 5–9, Riva del Garda, Italy.
- Raskin, J. (2000). *The Humane Interface. New directions for Designing Interactive Systems*. Addison Wesley, Boston, MA.
- Reithinger, N., Bergweiler, S., Engel, R., Herzog, G., Pfleger, N., Romanelli, M., and Sonntag, D. (2005a). A Look Under the Hood: Design and Development of the First Smartweb System Demonstrator. In *Proceedings of the 7th international conference on Multimodal interfaces (ICMI'05)*, pages 159–166, New York, NY. ACM.
- Reithinger, N., Fedeler, D., Kumar, A., Lauer, C., Pecourt, E., and Romary, L. (2005b). MIAMM: A Multi-Modal Dialogue System Using Haptics. In van Kuppevelt, J., Dybkjaer, L., and Bernsen, N., editors, *Advances in Natural Multimodal Dialogue Systems*, pages 307–332. Springer, Dordrecht, The Netherlands.
- Reithinger, N., Gebhard, P., Löckelt, M., Ndiaye, A., Pfleger, N., and Klesen, M. (2006). VirtualHuman—Dialogic and Emotional Interaction with Virtual Characters. In *Proceedings of the 8th International Conference on Multimodal Interfaces (ICMI'06)*, pages 51–58, Banff, Canada.
- Reithinger, N. and Sonntag, D. (2005). An Integration Framework for a Mobile Multimodal Dialogue System Accessing the Semantic Web. In *Proceedings of the 9th European Conference on Speech Communication and Technology (Interspeech/Eurospeech)*, pages 841–844, Lisbon, Portugal.
- Rich, C., Sidner, C. L., and Lesh, N. (2001). COLLAGEN: Applying Collaborative Discourse Theory to Human-Computer Interaction. *Artificial Intelligence Magazine*, 22(4):15–25.

- Romanelli, M., Becker, T., and Alexandersson, J. (2005). On Plurals and Overlay. In *Proceedings of the 9th Workshop on the Semantics and Pragmatics of Dialogue (Dialor)*, pages 101–108, Nancy, France.
- Roy, D., Hsiao, K.-Y., and Mavridis, N. (2004). Mental Imagery for a Conversational Robot. *IEEE Transactions on Systems, Man, and Cybernetics*, 34(3):1374–1383.
- Russell, S. and Norvig, P., editors (1995). *Artificial Intelligence. A Modern Approach*. Prentice Hall Series in Artificial Intelligence. Prentice Hall, Englewood Cliffs, NJ.
- Rutter, D. (1984). *Looking and Seeing. The Role of Visual Communication in Social Interaction*. John Wiley & Sons Ltd., Chichester, NY.
- Sacks, H., Schegloff, E. A., and Jefferson, G. (1974). A Simplest Systematics for the Organization of Turn-Taking for Conversations. *Language*, 50(4):696–734.
- Salmon-Alt, S. (2000). Interpreting Referring Expressions by Restructuring Context. In *Proceedings of ESSLLI 2000 (Student Session)*, Birmingham, UK.
- Sanchez-Nielsen, E., Anton-Canalis, L., and Hernandez-Tejera, M. (2004). Hand Gesture Recognition for Human-Machine Interaction. In *12th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, pages 395–403, Plzen-Bory, Czech Republic.
- Scherer, K. and Wallbott, H. (1985). Analysis of Nonverbal Behavior. In *Handbook of Discourse Analysis*, volume 2, pages 199–230. Academic Press, London, UK.
- Schlangen, D. (2003). *A Coherence-Based Approach to the Interpretation of Non-Sentential Utterances in Dialogue*. PhD thesis, School of Informatics, University of Edinburgh.
- Schlangen, D. and Lascarides, A. (2003). The Interpretation of Non-Sentential Utterances in Dialogue. In *Proceedings of the 4th SIGdial Workshop on Discourse and Dialogue*, Sapporo, Japan.
- Schmitt, M. (2005). *Dynamische Modellierung interpersoneller Beziehungen zwischen virtuellen Charakteren*. PhD thesis, Department of Computer Science, Saarland University.
- Schober, M. and Clark, H. H. (1989). Understanding by Addressee and Overhearers. *Cognitive Psychology*, 21(2):211–232.
- Searle, J. (1969). *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, Cambridge, UK.
- Searle, J. R. (1975). Indirect Speech Acts. In Cole, P. and Morgan, J., editors, *Syntax and Semantics*, volume 3. Academic Press, New York, NY.
- Seneff, S., Hurley, E., Lau, R., Pao, C., Schmid, P., and Zue, V. (1998). Galaxy-II: A Reference Architecture for Conversational System Development. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP'98)*, pages 931–934, Sydney, Australia.

- Sicconi, R., Potamianos, G., Libal, V., Huang, J., Marcheret, E., Chu, S., and Connell, J. (2005). A Prototype System for Audio-Visual Automatic Speech Recognition. In *Proceedings of the Demo Session of the 7th International Conference on Multimodal Interfaces (ICMI'05)*, pages 45–47, Trento, Italy.
- Siepmann, R., Batliner, A., and Oppermann, D. (2001). Using Prosodic Features to Characterize Off-Talk in Human-Computer Interaction. In *Proceedings of the ISCA Workshop on Prosody in Speech Recognition and Understanding*, pages 147–150, Red Bank, NJ.
- Skantze, G. (2006). GALATEA: A Discourse Modeller Supporting Concept-level Error Handling in Spoken Dialogue Systems. In *Proceedings of the 6th SIGDial Workshop on Discourse and Dialogue*, pages 178–189, Lisbon, Portugal.
- Smith, B. (2003). Ontology. In Floridi, L., editor, *Blackwell Guide to the Philosophy of Computing and Information*, pages 155–166. Blackwell, Oxford.
- Sonntag, D. (2006). Towards Combining Finite State, Ontologies, and Data Driven Approaches to Dialogue Management for Question Answering. In *Proceedings of the 5th Slovenian and First International Language Technology Conference (IS-LTC)*, pages 210–215, Ljubljana, Slovenia.
- Sonntag, D., Engel, R., Herzog, G., Pfalzgraf, A., Pflieger, N., Romanelli, M., and Reithinger, N. (2007a). SmartWeb Handheld–Multimodal Interaction with Ontological Knowledge Bases and Semantic Web Services. In *Proceedings of the International Workshop on AI for Human Computing (AI4HC'07)*, pages 115–122, Hyderabad, India.
- Sonntag, D., Engel, R., Herzog, G., Pfalzgraf, A., Pflieger, N., and Romanelli, Massimo Reithinger, N. (2007b). SmartWeb Handheld—Multimodal Interaction with Ontological Knowledge Bases and Semantic Web Services (extended version). In Huang, T., Nijholt, A., Pantic, M., and Plentland, A., editors, *Artificial Intelligence for Human Computing*, pages 272–295. Springer, Heidelberg.
- Sonntag, D. and Romanelli, M. (2006). A Multimodal Result Ontology for Integrated Semantic Web Dialogue Applications. In *Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06)*, pages 511–516, Genova, Italy.
- Sowa, T., Fröhlich, M., and Latoschik, M. E. (1999). Temporal Symbolic Integration Applied to a Multimodal System Using Gestures and Speech. In *Proceedings of the International Gesture Workshop*, pages 291–302, Berlin, Germany. Springer.
- Stalnaker, R. (1978). Assertions. In Cole, P., editor, *Syntax and Semantics*, volume 9, pages 315–332. Academic Press, New York, NY.
- Stenström, A.-B. (1994). *An Introduction to Spoken Interaction*. Longman Group, London, UK.
- Streit, M., Batliner, A., and Portele, T. (2006). Emotion Analysis and Emotion-Handling Subdialogues. In Wahlster, W., editor, *SmartKom—Foundations of Multimodal Dialogue Systems*, pages 317–332. Springer, Heidelberg, Germany.

- Streit, M. and Krieger, H.-U. (2004). Ellipsis Resolution by Controlled Default Unification for Multi-Modal and Speech-Dialog Systems. In Harabagiu, S. and Farwell, D., editors, *Proceedings of the ACL 2004 Workshop on Reference Resolution and Its Applications*, pages 87–94, Barcelona, Spain.
- Swartout, W., Hill, R., Gratch, J., Johnson, W., Kyriakakis, C., Labore, K., Lindheim, R., Marsella, S., Miraglia, D., Moore, B., Morie, J., Rickel, J., Thiebaux, M., Tuch, L., and Whitney, R. (2001). Toward the Holodeck: Integrating Graphics, Sound, Character and Story. In *Proceedings of 5th the International Conference on Autonomous Agents*, pages 409–416, New York, NY. ACM.
- ten Bosch, L., Oostdijk, N., and de Ruiter, J. P. (2004). Durational Aspects of Turn-Taking in Spontaneous Face-to-Face and Telephone Dialogues. In *Proceedings of the Conference on Text, Speech and Dialogue*, pages 563–570, Heidelberg, Germany. Springer.
- Tepperman, J., Traum, D., and Narayanan, S. (2006). Yeah Right: Sarcasm Recognition for Spoken Dialogue Systems. In *Proceedings of the International Conference on Spoken Language Processing (Interspeech/ICSLP)*, pages 1838–1841, Pittsburgh, PA.
- Thórisson, K. R. (1994). Face-to-Face Communication with Computer Agents. In *AAAI Spring Symposium on Believable Agents Working Notes*, pages 18–22. Stanford University, California.
- Thórisson, K. R. (1996). *Communicative Humanoids: A Computational Model of Psychosocial Dialogue Skills*. PhD thesis, Massachusetts Institute of Technology.
- Thórisson, K. R. (1997). Gandalf: An Embodied Humanoid Capable of Real-Time Multimodal Dialogue with People. In *Proceedings of the first ACM International Conference on Autonomous Agents*, pages 536–537, Marina del Rey, CA.
- Thórisson, K. R. (1999). A Mind Model for Multimodal Communicative Creatures and Humanoids. *International Journal of Applied Artificial Intelligence*, 13(4-5):449–486.
- Thórisson, K. R. (2002). Natural Turn-Taking Needs No Manual: Computational Theory and Model, from Perception to Action. In B. Granström, D. House, I. K., editor, *Multimodality in Language and Speech Systems*, pages 173–207. Kluwer Academic Publishers, Dordrecht, The Netherlands.
- Tomabechi, H. (1992). Quasi-Destructive Graph Unification with Structure-Sharing. In *Proceedings of the 14th International Conference on Computational Linguistics*, pages 440–446, Nantes, France.
- Tondering, C. (2005). Frequently Asked Questions about Calendars (Version 2.8). (Online; last accessed 05.01.2007, <http://www.tondering.dk/claus/cal/calendar28.pdf>).
- Traum, D. (1994). *A Computational Theory of Grounding in Natural Language Conversation*. PhD thesis, Department of Computer Science, University Rochester.
- Traum, D. (2004). Issues in Multi-party Dialogues. In Dignum, F., editor, *Advances in Agent Communication*, LNAI 2922, pages 201–211. Springer, Heidelberg, Germany.

- Traum, D. and Allen, J. (137–140). A Speech Acts Approach to Grounding in Conversation. In *Proceedings of the 2nd International Conference on Spoken Language Processing (ICSLP'92)*, Banff, Canada.
- Traum, D. R. (1999). Computational Models of Grounding in Collaborative Systems. In *Working Notes of AAAI Fall Symposium on Psychological Models of Communication*, pages 124–131, North Falmouth, MA.
- Traum, D. R. (2000). 20 Questions for Dialogue Act Taxonomies. *Journal of Semantics*, 17(1):7–30.
- Traum, D. R. and Hinkelman, E. A. (1992). Conversation Acts in Task-Oriented Spoken Dialogue. *Computational Intelligence*, 8(3):575–599.
- Traum, D. R. and Rickel, J. (2002). Embodied Agents for Multi-party Dialogue in Immersive Virtual Worlds. In *Proceedings of the International Conference on Autonomous Agents (AAMAS 2002)*, pages 766–773, Bologna, Italy.
- Vertegaal, R., Slagter, R., van der Veer, G., and Nijholt, A. (2001). Eye Gaze Patterns in Conversations: There is More to Conversational Agents Than Meets the Eyes. In Jacko, J., Sears, A., Beaudouin-Lafon, M., and Jacob, R., editors, *Proceedings of the Conference on Human Factors and Computing Systems (CHI'01)*, pages 301–308, Seattle, WA. ACM.
- Vuurpijl, L., ten Bosch, L., Rossignol, S., Neumann, A., Engel, R., and Pflieger, N. (2004). COMIC Deliverable 3.3: Reports on Human Factors Experiments with Simultaneous Coordinated Speech and Pen Input and Fusion. Technical report, The COMIC Project.
- Wahlster, W. (1988). Distinguishing User Models from Discourse Models. *Computational Linguistics*, 14(3):101–113.
- Wahlster, W. (1991). User and Discourse Models for Multimodal Communication. In J. W. Sullivan and S. W. Tyler, editor, *Intelligent User Interfaces*, pages 45–67. ACM.
- Wahlster, W. (2002). Disambiguierung durch Wissensfusion: Grundprinzipien der Sprachtechnologie. *KI—Künstliche Intelligenz*, 16(1):48–50.
- Wahlster, W. (2003a). SmartKom: Symmetric Multimodality in an Adaptive and Reusable Dialogue Shell. In Krahl, R. and Günther, D., editors, *Proceedings of the Human Computer Interaction Status Conference 2003*, pages 47–62, Berlin: DLR.
- Wahlster, W. (2003b). Towards Symmetric Multimodality: Fusion and Fission of Speech, Gesture, and Facial Expression. In Günter, A., Kruse, R., and Neumann, B., editors, *KI 2003: Advances in Artificial Intelligence*, pages 1–18, Berlin, Heidelberg, Germany. Springer.
- Wahlster, W. (2004a). Conversational User Interfaces. *it: Information Technology*, 46(6):289–290.
- Wahlster, W. (2004b). SmartWeb: Mobile Applications of the Semantic Web. In Dadam, P. and Reichert, M., editors, *GI Jahrestagung*, pages 26–27, Heidelberg, Germany. Springer.

- Wahlster, W. (2006). *SmartKom—Foundations of Multimodal Dialogue Systems*. Springer, Heidelberg, Germany.
- Wahlster, W. and Kobsa, A. (1989). User Models in Dialog Systems. In Kobsa, A. and Wahlster, W., editors, *User Models in Dialog Systems*, pages 4–34. Springer, Berlin, Germany.
- Wahlster, W., Reithinger, N., and Blocher, A. (2001). SmartKom: Multimodal Communication with a Life-Like Character. In *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech'01)*, volume 3, pages 1547–1550, Aalborg, Denmark.
- Walton, D. N. (1992). Types of Dialogue, Dialectical Shifts, Fallacies. In van Eemeren, F. H., Grootendorst, R., Blair, J. A., and Willard, C. A., editors, *Argumentation Illuminated*, pages 133–147. International Centre for the Study of Argumentation (SICSAT), Amsterdam.
- Ward, N. (2000). Issues in the Transcription of English Conversational Grunts. In *Proceedings of the 1st SIGDial Workshop on Discourse and Dialogue*, pages 29–35, Hong Kong, China.
- Wasinger, R. (2006). *Multimodal Interaction with Mobile Devices: Fusing a Broad Spectrum of Modality Combinations*. Dissertationen zur Künstlichen Intelligenz (DISKI). Akademische Verlags Gesellschaft, Berlin, Germany.
- Wasinger, R., Krüger, A., and Jacobs, O. (2005). Integrating Intra and Extra Gestures into a Mobile and Multimodal Shopping Assistant. In *Proceedings of the 3rd International Conference on Pervasive Computing (Pervasive)*, pages 297–314, Munich, Germany.
- Wasinger, R. and Wahlster, W. (2006). The Anthropomorphized Product Shelf: Symmetric Multimodal Interaction with Instrumented Environments. In Aarts, E. and Encarnacao, J., editors, *True Visions: The Emergence of Ambient Intelligence*, pages 291–306. Springer, Heidelberg, Germany.
- Webber, B. (1991). Structure and Ostension in the Interpretation of Discourse Deixis. *Natural Language and Cognitive Processes*, 6(2):107–135.
- Weischedel, R. M. and Sondheimer, N. K. (1982). An Improved Heuristic for Ellipsis Processing. In *Proceedings of the 20th Annual Meeting on Association for Computational Linguistics (ACL'82)*, pages 85–88, Toronto, Canada.
- Winograd, T. (1972). *Understanding Natural Language*. Academic Press, Edingburgh, UK.
- Woods, W. (1978). Semantics and Quantification in Natural Language Question Answering. In Yovits, M., editor, *Advances in Computers*, volume 17, pages 2–64. Academic Press, New York, NY.
- Wu, L., Oviatt, S. L., and Cohen, P. R. (1999). Multimodal Integration—A Statistical View. *IEEE Transactions on Multimedia*, 1(4):334–341.
- Yngve, V. H. (1970). On Getting a Word in Edgewise. In *Papers from the 6th Regional Meeting of the Chicago Linguistics Society*, pages 567–577. Chicago Linguistics Society.