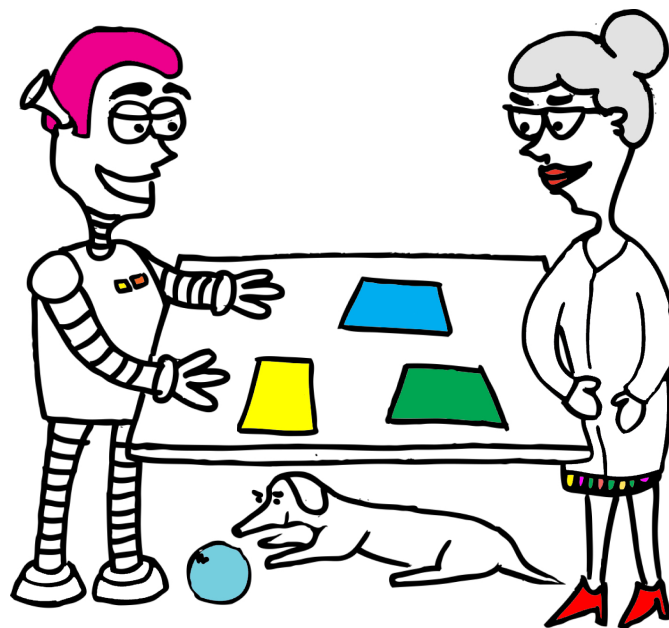




MODELING

INTERPERSONAL COORDINATION AND GROUNDING BEHAVIOR

IN JOINT ACTIVITIES WITH SOCIAL COMPANIONS



DISSERTATION

ZUR ERLANGUNG DES AKADEMISCHEN GRADES

DOKTOR DER INGENIEURWISSENSCHAFTEN

EINGEREICHT AN DER
FAKULTÄT FÜR ANGEWANDTE INFORMATIK
UNIVERSITÄT AUGSBURG

VON

GREGOR ULRICH MEHLMANN
AUGSBURG, DEUTSCHLAND, 2018

GUTACHTER:

PROF. DR. ELISABETH ANDRÉ

PROF. DR. BERNHARD MÖLLER

PROF. DR. WOLFGANG MINKER

PRÜFUNGSdatum: 04.04.2018

ZUSAMMENFASSUNG

Zwischenmenschliche Koordination und Grounding sind zwei faszinierende Phänomene, die allen natürlichen, sozialen Interaktionen und alltäglichen gemeinsamen Aktivitäten zwischen uns Menschen zugrunde liegen. Zwischenmenschliche Koordination findet immer dann statt, wenn sich Menschen während einer Interaktion wechselseitig und reibungslos an den Rhythmus und das Tempo des Gegenübers anpassen oder ihre Handlungen und Verhaltensweisen synchronisieren und nahtlos miteinander verzahnen und verknüpfen. Grounding bezeichnet alle konstanten, gemeinschaftlichen Bemühungen, die mit der Herstellung, Aufrechterhaltung und Wiederherstellung eines gemeinsamen Wahrnehmungsbereiches und Gesprächshintergrunds während solchen sozialen Interaktionen verbunden sind.

Zwischenmenschliche Koordination und Grounding werden durch eine Vielzahl miteinander verwobener, teilweise konkurrierender, Verhaltensfunktionen beeinflusst, welche unterschiedliche soziale und regulative Aufgaben haben und zahlreiche Verhaltensmodalitäten umfassen. So spielen verschiedene Blickverhalten wichtige Rollen beim Lenken und Folgen von Aufmerksamkeit, der multimodalen Disambiguierung von Referenzausdrücken, dem Aushandeln der Rederechtvergabe, der Erzeugung von Rückmeldungen, der Regulierung der Intimität, sowie der Bewertung von Persönlichkeit. Ein weiterer wichtiger Aspekt sind Sprachüberlappungen und Unterbrechungsversuche, die einen deutlichen Einfluss auf den Informations- und Interaktionsfluss haben und soziale Haltungen sowie zwischenmenschliche Beziehungen, wie Engagement, Dominanz oder Zugehörigkeitsgefühl signalisieren.

Die enge Abstimmung dieser Verhaltensfunktionen und deren Zusammenspiel mit dem Dialogmanagement zu meistern, spielt eine wichtige Rolle für zwischenmenschliche Koordination und Grounding in physisch situierten, gemeinsamen Aktivitäten. Dies beinhaltet hauptsächlich die geschickte Synchronisation, das heißt die richtige Priorisierung und reziproke Verzahnung der zugrunde liegenden, nebenläufigen Verhaltens- und Berechnungsprozesse. Während dies bei natürlichen, menschlichen Interaktionen schon von Geburt an so reibungslos und intuitiv zu funktionieren scheint, geschieht dies in sozialen Interaktionen zwischen Menschen und sozialen Agenten noch keineswegs so perfekt. Kleinste Diskrepanzen bei der Synchronisation von Verhaltensaspekten können einen sozialen Begleiter unnatürlich, inkompetent oder ungeschickt erscheinen lassen. Aus diesem Grund ist die enge Koordination der Verhaltensfunktionen und der zugrunde liegenden Prozesse eine der größten Herausforderung bei der Modellierung des Interaktionsverhaltens von sozialen Agenten.

Um dieser Modellierungsherausforderung gerecht zu werden, muss ein Autor drei Modellierungsaufgaben bewältigen, die jeweils aufgabenspezifische Anforderungen für einen ausreichend ausdrucksstarken und praktikablen Modellierungsansatz mit sich bringen. Die erste Aufgabe beinhaltet die Erzeugung von vielseitigen, multimodalen Verhaltenskompositionen, die Kontextwissen integrieren und automatisch variiert werden können. Die zweite Aufgabe umfasst die Auswertung von zeitlichen und semantischen Bedingungen sowie Quantifikationen bei der multimodalen Fusion und Schlussfolgerung auf Wissen. Die dritte Aufgabe ist die inkrementelle Verzahnung von Verhaltenserkennung, Wissensargumentation und Verhaltensgenerierung sowie die Unterbrechung und Wiederaufnahme von nebenläufigen, verschachtelten und eng miteinander verflochtenen Prozessen im Verhaltensmodell.

Eine Überprüfung der aktuellen, verwandten Arbeiten zeigt, dass bisher kein einheitliches Verhaltens- und Interaktionsmodellierungskonzept für soziale Begleiter vorgestellt wurde, welches alle diese Aufgaben und Anforderungen bewältigt. Einige verwandte Arbeiten haben einzelne Verhaltensfunktionen oder Aspekte isoliert betrachtet und modelliert, ohne ihr komplexes Zusammenspiel mit anderen Funktionen oder ihre Rolle für die zwischenmenschliche Koordination und das Grounding zu berücksichtigen. Andere Forschung konzentriert sich auf die Entwicklung von allzwecklichen Modellierungssprachen für einzelne der Modellierungsaufgaben, vermisst aber völlig, zu demonstrieren, ob und wie man ihre Lösung mit den anderen Aufgaben verknüpfen kann und inwiefern man diese für die Modellierung von zwischenmenschlicher Koordination und Grounding verwenden kann.

Die vorliegende Arbeit stellt einen neuartigen Lösungsansatz für die Modellierung des Interaktionsverhaltens künstlich und sozial intelligenter Agenten vor. Dieser ist genau darauf ausgelegt, die genannten Anforderungen zu erfüllen und damit die Unzulänglichkeiten verschiedener verwandter Ansätze zu überwinden. Es ist der erste solche Ansatz, der eine speziell entworfene, hierarchische und parallele State-Chart-Variante, einen domänenspezifischen, logischen, multimodalen Fusions- und Argumentationskalkül, und eine vorlagenbasierte Verhaltensbeschreibungssprache zu einem einheitlichen Ansatz zur Verhaltens- und Interaktionsmodellierung vereint. Dieser Ansatz erleichtert die verteilte und iterative Entwicklung von klar strukturierten, leicht anpassbaren, erweiterbaren und wiederverwendbaren Modellen für das Dialog- und Interaktionsverhalten von sozialen Agenten. Der Ansatz ist ausreichend ausdrucksstark aber gleichzeitig bemerkenswert praktikabel, da er hauptsächlich auf visuellen und deklarativen Formalismen zur Modellierung beruht. Er eignet sich somit sowohl für die schnelle Entwicklung von einfachen Prototypen als auch für die schrittweise Erstellung recht anspruchsvoller Modelle durch Autoren mit unterschiedlichen Erfahrungen, Expertisen und Herangehensweisen.

ABSTRACT

Interpersonal coordination and grounding are fascinating phenomena that underlie all natural social interactions and everyday joint activities between humans. Interpersonal coordination takes place whenever people are smoothly adapting to each other's interaction tempo and rhythm, or are seamlessly synchronizing and intertwining their actions and behaviors. Grounding denotes all collaborative efforts that are involved in establishing, maintaining, and repairing the common perceptual and conversational ground during a joint activity.

*INTERPERSONAL
COORDINATION
& GROUNDING*

Interpersonal coordination and grounding are influenced by a variety of highly interwoven behavioral functions that have different social and regulative impacts and include numerous information modalities. For example, various gaze behaviors are involved in attention following, multi-modal disambiguation, turn management, feedback generation, intimacy regulation, and personality evaluation. Another important aspect is speech overlaps and interruption attempts that have a significant effect on the interaction flow as well as interpersonal attitudes, such as involvement, dominance, and affiliation.

*BEHAVIORAL
FUNCTIONS*

Mastering the close coordination of the above functions and their interplay with the dialog management plays an important role for interpersonal coordination and grounding in physically situated joint activities. This mainly comprises the proper prioritization and tight synchronization of the underlying incremental, reciprocal, and concurrent behavioral processes. While this seems to work so well and intuitively in natural human interactions, already from birth on, it obviously works by no means as perfectly in social human-agent interactions. Smallest discrepancies in the synchronization or prioritization of individual behavioral aspects may already make a social companion appear unnatural, incompetent, or clumsy. For that reason, the close coordination of the behavioral functions and the underlying behavioral processes is the major challenge in modeling the behavior and interaction of a social agent.

*MODELING
CHALLENGE*

In order to meet this modeling challenge, an author must cope with three modeling tasks, each of which establishing task-specific requirements for an sufficiently expressive and practicable modeling approach. The first task involves the creation of versatile, multi-modal behavior compositions that integrate context knowledge and can automatically be varied. The second task includes the evaluation of temporal and semantic integration constraints as well as various quantification operations for multi-modal fusion and knowledge reasoning. The last task comprises the proper incremental interleaving of behavior recognition, knowledge reasoning and behavior generation as well as the close coordination, prioritization, inter-

*MODELING
REQUIREMENTS*

ruption, and resumption of concurrent, nested, and intertwined processes underlying the various behavioral functions on different levels of the behavior model.

RELATED EFFORTS A review of state-of-the-art related efforts shows that, so far, there has not been presented a uniform behavior and interaction modeling approach for social companions that masters all of these tasks and requirements. Some related work has studied and modeled individual behavioral functions or aspects in isolation without considering their complex interplay with other functions or their role for interpersonal coordination and grounding. Other research has focused on the design of general purpose modeling languages for only one individual modeling tasks but misses to demonstrate how to interlink their solution with the other tasks and how to use them for modeling interpersonal coordination and grounding behaviors.

MODELING APPROACH This thesis presents a novel modeling approach for the interactive behavior of artificially and socially intelligent agents that is precisely designed to meet the above requirements and overcome the shortcomings of related work. It is the first approach to combine a specially designed, hierarchical and concurrent state-chart variant, a domain-specific, logic, multi-modal fusion and reasoning calculus, and a template-based behavior description language into a uniform behavior and interaction modeling framework. The proposed approach significantly facilitates the distributed and iterative development of clearly structured, easily adaptable, extensible, and reusable computational dialog, behavior, and interaction models of social agents. While being sufficiently expressive, it is remarkably practicable since it primarily relies on visual and declarative modeling formalisms. It is suitable for rapid-prototyping and the creation of sophisticated models by differently experienced authors.

DANKSAGUNG

An erster Stelle bedanke ich mich bei meiner Betreuerin, Frau Prof. Dr. Elisabeth André, für die Möglichkeit an ihrem Lehrstuhl arbeiten, lehren, forschen und lernen zu dürfen. Ich danke ihr für die hervorragende Unterstützung meiner Forschung und die Betreuung dieser Doktorarbeit während der letzten Jahre. Mein Dank geht außerdem an meine beiden Zweitgutachter, Prof. Dr. Bernhard Möller und Prof. Dr. Wolfgang Minker, die diese Arbeit mit ihren sehr nützlichen Hinweisen und Vorschlägen unterstützt haben.

Dank gilt auch meinen vielen Kolleginnen und Kollegen mit denen ich zusammenarbeiten durfte. Gerne erinnere ich mich an die erfolgreiche Zusammenarbeit, freundschaftliche Arbeitsatmosphäre und viele interessante Diskussionen in den letzten Jahren. Allen voran danke ich meinem geschätzten Kollegen Dr. Patrick Gebhard mit dem ich seit vielen Jahren sehr erfolgreich zusammen arbeiten darf. Er hat maßgeblich zum Erfolg dieser Arbeit beigetragen — auch weil er mich stets aufmunterte wenn ich daran gezweifelt habe.

Ein ganz besonderer Dank gilt meiner Familie, meinen lieben Eltern und Brüdern, die mir in schwierigen Momenten eine emotionale Stütze waren und immer wieder den notwendigen Ehrgeiz in mir geweckt haben. Für die Liebe und Unterstützung meiner Eltern, Dr. Hiltrud Mehlmann und Dr. Ulrich Mehlmann, bin ich unendlich dankbar. Ich wünschte mein Vater könnte den erfolgreichen Abschluss der Arbeit nun zusammen mit uns feiern.

Zuletzt gilt mein Dank vor allem meiner Partnerin Claudia, die immer wieder mit großer Toleranz und Geduld meine Launen und Beschwerden ertragen hat und zu meinem Erstaunen immer noch mit mir zusammen ist. Außerdem danke ich meinen geliebten Söhnen, Karl und Paul, die mich immer wieder erfolgreich von der Arbeit abzulenken wussten.

*“Nicht die Glücklichen sind dankbar —
Es sind die Dankbaren, die glücklich sind.”*

FRANCIS BACON

CONTENTS

List of Figures	V
List of Listings	IX
I Introduction and Background	1
1 Introduction — Joint Activities with Social Companions	3
1.1 General Motivation	4
1.1.1 Joint Activities	5
1.1.2 Social Companions	6
1.2 Introductory Scenario	7
1.2.1 Interaction Extract	8
1.2.2 Scenario Discussion	9
1.3 Research Objectives	10
1.3.1 Coordinating Functions and Processes	10
1.3.2 Integrating Input and Context Events	11
1.3.3 Creating Behavior and Dialog Content	11
1.4 Thesis Organization	11
1.4.1 Scientific Approach	12
1.4.2 Thesis Structuring	12
2 Background — Interpersonal Coordination and Grounding	15
2.1 Interpersonal Coordination and Grounding	16
2.1.1 Interpersonal Coordination	16
2.1.2 Grounding in Joint Activity	22
2.1.3 Synergy and Overlap Effects	28
2.2 The Different Functions of Gaze Behavior	30
2.2.1 Concepts and Definitions	30
2.2.2 Attention and Intention	33
2.2.3 Understanding and Recall	34
2.2.4 Turn-Taking and Feedback	35
2.2.5 Emotions and Cognition	36
2.2.6 Personality and Intimacy	38

2.3	The Functions of Overlaps and Interruptions	39
2.3.1	Concepts and Definitions	39
2.3.2	Turn-Taking and Feedback	42
2.3.3	Attitudes and Relationships	43
2.4	Summary and Conclusion	45
II Challenges and Related Work		47
3	Challenges — Modeling Subtasks, Requirements and Solutions	49
3.1	Subtasks, Requirements and Solutions	50
3.2	Coordinating Functions and Processes	54
3.2.1	Incremental and Reciprocal Meshing	54
3.2.2	Parallel and Hierarchical Structuring	56
3.2.3	Interruption and Coherent Resumption	59
3.3	Integrating Input and Context Events	61
3.3.1	Uniform Knowledge Representation	62
3.3.2	Well-Organized Working Memory	63
3.3.3	Multi-Modal Fusion and Reasoning	65
3.4	Creating Behavior and Dialog Content	67
3.4.1	Versatile Composition of Behavior	68
3.4.2	Flexible Integration of Knowledge	69
3.4.3	Automatic Variability of Behavior	70
3.5	Summary and Conclusion	72
4	Related Work — Behavioral Aspects and Modeling Approaches	73
4.1	Modeling Behavioral Functions	74
4.1.1	Modeling Functions of Gaze Behavior	74
4.1.2	Modeling Functions of Voice Overlaps	79
4.2	Modeling Tasks and Requirements	80
4.2.1	Coordinating Functions and Processes	80
4.2.2	Integrating Input and Context Events	88
4.2.3	Creating Behavior and Dialog Content	96
4.3	Summary and Conclusion	99
III Conception and Illustration		101
5	Conception — Designing the Behavior Flow Modeling Language	103
5.1	Guidelines, Conditions and Approach	104
5.2	Creating Behavior and Dialog Content	109
5.2.1	Behavioral Activity Specification	109
5.2.2	Parameterization and Variation	115
5.3	Integrating Input and Context Events	118

5.3.1	Feature Structure Representation	118
5.3.2	Logic Fact Base and Event History	121
5.3.3	Multi-Modal Fusion and Reasoning	126
5.4	Coordinating Functions and Processes	133
5.4.1	States, Transitions and Variables	133
5.4.2	Decomposition and Synchronization	141
5.4.3	Interruption and History Mechanism	146
5.5	Summary and Conclusion	149
6	Illustration — Modeling Social and Collaborative Joint Activities	151
6.1	Setup, Architecture and Scenarios	152
6.1.1	System Setup And Architecture	152
6.1.2	Interaction Scenario Description	154
6.2	Preprocessing Input and Context	157
6.2.1	Representing Domain Knowledge	157
6.2.2	Preprocessing User Input Events	158
6.2.3	Disambiguating Speech with Gaze	162
6.3	Modeling Behavior and Interaction	165
6.3.1	Basic Input Event Handling	166
6.3.2	Behavioral Pattern Recognition	170
6.3.3	Participant Role Management	183
6.3.4	Dialog and Behavior Control	189
6.4	Summary and Conclusion	195
IV	Realization and Conclusion	197
7	Realization — Re-engineering and Validating the VSM³ Framework	199
7.1	Redefining Language Specifications	200
7.1.1	The Specification of <i>SFSCs</i>	201
7.1.2	The Specification of <i>SFGL</i>	205
7.1.3	The Specification of <i>SFSL</i>	211
7.1.4	The Integration of <i>SFQL</i>	215
7.2	Refactoring Software Components	215
7.2.1	Design Considerations	216
7.2.2	Data Model Definition	219
7.2.3	Runtime Environment	222
7.2.4	Modeling Environment	226
7.3	Developing Demo Applications	229
7.3.1	Agents in a Virtual School Yard	229
7.3.2	Actors in a Virtual Soap Opera	231
7.3.3	Coaches in Interview Trainings	233
7.3.4	Helpers in Shared Workspaces	235

7.3.5	Further Applications with <i>VSM</i> ³	237
7.4	Summary and Conclusion	240
8	Conclusion — Summary, Contributions and Future Work	241
8.1	Contributions	242
8.1.1	Scientific Approach and Footing	242
8.1.2	The <i>BFML</i> Modeling Framework	242
8.1.3	The <i>VSM</i> ³ Authoring Software	244
8.2	Future Work	244
8.2.1	Corpus-Based Model Refinement	245
8.2.2	Data-Driven Behavior Adaptation	245
	Bibliography	247

LIST OF FIGURES

1.2.1	The physically situated, collaborative joint activity between <i>Charly</i> (A) and <i>Marley</i> (B).	7
1.4.1	The overview of the scientific approach and the structuring of this dissertation.	12
2.2.1	An illustration of some of the most important gaze mechanisms in social interactions.	31
2.2.2	An illustration of some of the most important participant roles in social interactions.	32
2.3.1	The types of speaker switch attempts that Beattie (1981a) adopted from Ferguson (1977).	40
2.3.2	The types of simultaneous speech events as presented in Roger and Schumacher (1983).	41
3.2.1	An illustration of the parallel and hierarchical decomposition of the behavior model.	57
3.3.1	An illustration of the delays caused by the processing times of speech and gaze inputs.	64
3.3.2	An illustration of a quantification of gaze events related to a verbal reference expression.	67
5.1.1	A white board used to sketch a model with states, transitions, variables and events.	105
5.1.2	The architecture of the <i>BFML</i> modeling language ensemble developed in this thesis.	107
5.2.1	The syntax of an <i>action activity</i> specifying an <i>actor</i> and <i>action</i> as well as a <i>feature list</i> .	110
5.2.2	Some examples of action activities that specify nonverbal behaviors of the agent <i>Charly</i> .	110
5.2.3	Examples of action activities that execute application- and device-specific commands.	110
5.2.4	The syntax of an <i>utterance activity</i> with an <i>actor</i> , the <i>content</i> , and a <i>punctuation mark</i> .	111
5.2.5	Some examples of multi-modal <i>utterance activities</i> containing <i>nested action activities</i> .	112
5.2.6	Some examples of multi-modal utterance activities containing nested prosodic activities.	112
5.2.7	The syntax of a <i>scene activity</i> with the <i>scene header</i> (A) and the <i>scene body</i> (B).	112
5.2.8	An example for the specification of a scene activity with the two agents <i>Charly</i> and <i>Reeti</i> .	113
5.2.9	Some examples of <i>blocking</i> and <i>non-blocking</i> playback calls for different activities.	114
5.2.10	An example of an activity playback command that executes the scene from Figure 5.2.8.	114
5.2.11	The command that is called to achieve the abortion of an agent's current behavior.	115
5.2.12	A scene definition with <i>placeholders</i> (A) and a call to this scene with <i>arguments</i> (B).	116
5.2.13	Some examples of <i>inline value insertion</i> when calling action and utterance activities.	117
5.2.14	A scene group with two different variations (A, B) of a scene with the same name.	118
5.3.1	The representation of a feature structure in graph-notation (A) and matrix-notation (B).	120
5.3.2	An example for the use of the logic matching operation predicate <i>val/3</i> in a query.	122
5.3.3	An example for the use of the logic modification operation predicate <i>set/4</i> .	123
5.3.4	Exemplary feature structures that are representing input events produced by the user.	124
5.3.5	Some pictorial illustrations of different qualitative temporal relations between events.	128
5.3.6	The exemplary use of the quantitative temporal relation predicate <i>rel_dist/3</i> in a query.	129
5.3.7	An example how a natural language quantification can be translated into a logic query.	131
5.3.8	An exemplary usage of the generalized quantifiers <i>forfraction/4</i> and <i>forlargest/3</i> .	132
5.4.1	A basic node (A) labeled with a variable definition (B) and a playback command (C).	134

5.4.2	A super node (A) defining local variables (B) and containing a short dialog phase with a single question-answer pair between the agent and the user in a nested subnode of the BFSC (C-F).	135
5.4.3	Two nodes connected with an <i>epsilon edge</i> that defines an unconditional transition.	136
5.4.4	Two nodes connected with a <i>timeout edge</i> that defines a scheduled transition.	137
5.4.5	Nodes connected with a set of <i>probability edges</i> that define probabilistic transitions.	137
5.4.6	Nodes connected with different <i>condition edges</i> that define conditional transitions.	138
5.4.7	Nodes connected with different <i>condition edges</i> that are labeled with logic queries.	138
5.4.8	An exemplary BFSC with transitions that have different evaluation policies.	139
5.4.9	A super node containing several parallel nested BFSCs with their start nodes.	143
5.4.10	Several nodes connected with a forking construct that is splitting the execution.	143
5.4.11	The usage of a shared variable named <i>sync</i> for the synchronization of BFSCs.	144
5.4.12	The usage of the built-in state condition <i>in/1</i> for the synchronization of BFSCs.	145
5.4.13	A signal exchange via the logic fact base using the predicates <i>signal/2</i> and <i>detect/2</i> .	146
5.4.14	The exemplary use of the history node and a history condition in a super node.	148
6.1.1	The general system setup of the applications that have been developed in this thesis.	152
6.1.2	A schematic illustration of the data flow within our application's software architecture.	153
6.1.3	The scenario and setup of the <i>ROBOTPUZZLE</i> application on the left and a capture of the eye-tracking video that has been processed in the <i>SSI</i> framework (Wagner <i>et al.</i> , 2013) on the right.	155
6.1.4	The setup of the <i>SOCIALCOACH</i> application with the user, the sensor setup, the virtual embodied characters as well as the <i>SSI</i> pipeline visualization and event monitor (Wagner <i>et al.</i> , 2013).	156
6.2.1	The exemplary representation of a puzzle piece (A) and a field (B) as feature structures.	157
6.2.2	The exemplary representation of an ambiguous (A) and an unambiguous instruction (B).	158
6.2.3	Examples of simple input events, like state (A), voice (B) and facial expression events (C).	158
6.2.4	The exemplary representation of a start touch (A) and a drag continue touch event (B).	159
6.2.5	The exemplary representation of a gaze distribution (A) and a gaze target event (B).	160
6.2.6	The exemplary representation of a set (A) and a choice question (B) speech act.	161
6.2.7	The exemplary representation of a check question without (A) and with spatial deixis (B).	162
6.2.8	An illustration of how the <i>forlargest/3</i> quantifier supports the disambiguation.	164
6.3.1	The overall architecture of the behavior flow model used in the illustrative application.	165
6.3.2	The behavior flow handling the user's and the agent's incoming voice activity events.	167
6.3.3	The behavior flow used for preprocessing the two participants' new gaze events.	168
6.3.4	The behavior flow used for handling touch events received from the surface table.	169
6.3.5	The behavior flow used for detecting and disambiguating the user's speech events.	169
6.3.6	The behavior flow used for recognizing the user's different turn-taking actions.	172
6.3.7	The behavior flow used for recognizing overlapping states and turn-taking conflicts.	173
6.3.8	The behavior flow used for recognizing different directions of voice activity overlaps.	174
6.3.9	The behavior flow used for recognizing the different types of voice overlap conflicts.	175
6.3.10	The behavior flow used for recognizing different gaze relations and connection events.	176
6.3.11	The behavior flow used for monitoring gaze relations between the user and the agent.	177
6.3.12	The behavior flow used for recognizing mutual facial gaze and directed gaze events.	178
6.3.13	The behavior flow used for recognizing back-channels, adjacency pairs and responses.	178
6.3.14	The behavior flow used for recognizing the user's verbal back-channel statements.	179
6.3.15	The behavior flow used for recognizing the user's nonverbal back-channel behaviors.	180
6.3.16	The behavior flow used for recognizing adjacency pair events or direct response events.	181

6.3.17	The behavior flow used for recognizing the different types of feedback eliciting cues. . . .	182
6.3.18	The behavior flow interpreting the user's gaze glances as back-channel eliciting cues. . . .	182
6.3.19	The behavior flow which is used for recognizing the user's facial mimicry eliciting cues. . .	183
6.3.20	The behavior flow managing the participant role assignment and shifting based on turn regulation signals from other layers of the model.	186
6.3.21	The behavior flow used for handling turn-taking and dialog signals as addressee.	187
6.3.22	The behavior flow used for used for handling turn-taking and dialog signals as speaker. . .	188
6.3.23	The behavior flow used for controlling the dialog flow and other behavioral aspects. . . .	189
6.3.24	The behavior flow used for coordinating the dialog flow with the dialog planning.	191
6.3.25	The behavior flow used for executing the agent's new contributions to the dialog.	192
6.3.26	The behavior flow controlling the agent's role-specific nonverbal behavior based on the signals from the lower layers of the model.	194
7.1.1	The architecture of the scene flow modeling language ensemble developed in this thesis. .	200
7.1.2	The visual syntax of a basic node with a few local definitions and command statements. .	202
7.1.3	The visual syntax of a super node with a history node, regular and alternative start nodes.	202
7.1.4	An exemplary <i>epsilon edge</i> which is connecting the source node N_1 with target node N_2 . .	203
7.1.5	An exemplary <i>timeout edge</i> which is connecting the source node N_1 with target node N_2 . .	203
7.1.6	Two exemplary <i>probability edges</i> from the source node N_1 to the target nodes N_2 and N_3 . .	204
7.1.7	An exemplary <i>condition edge</i> that is connecting the source node N_1 with target node N_2 . .	204
7.1.8	An exemplary <i>interruptive edge</i> connecting the source node N_1 with the target node N_2 . . .	204
7.1.9	Exemplary <i>forking edges</i> connecting the source node N_1 with target nodes N_2 and N_3	205
7.1.10	An exemplary scene script containing a comment and a single scene definition in <i>SFSL</i> . . .	211
7.1.11	A diagram showing the <i>SWI-PROLOG</i> module structure and relationships of the <i>SFQL</i>	215
7.2.1	A diagram showing an overview of important software layers and components of <i>VSM³</i> . . .	216
7.2.2	The sequence of steps performed by an author in order to finally execute a <i>SFML</i> model. . .	217
7.2.3	A diagram showing an extract of the data model definition on the data model layer.	219
7.2.4	A diagram partly showing the hierarchy of classes representing the <i>SFSC</i> constructs. . . .	220
7.2.5	A diagram partly showing the hierarchy of classes representing the <i>SFSL</i> elements.	220
7.2.6	A diagram partly showing the hierarchy of classes representing the <i>SFSC</i> constructs. . . .	221
7.2.7	A diagram showing some important classes of the interpreter runtime environment.	222
7.2.8	A diagram showing the relationships between the classes used for activity playback.	223
7.2.9	A diagram showing the relationships between the classes used for query execution.	225
7.2.10	A diagram showing important classes of the integrated development environment.	227
7.2.11	A screenshot of the <i>IDE</i> of <i>VSM³</i> showing the authoring suite's most important editor components.	228
7.3.1	Some hamster characters playing different pedagogical roles in the <i>DYNALearn</i> project. . . .	230
7.3.2	A use case with the teacher character explaining relations in a conceptual model.	230
7.3.3	A use case with the quiz master character playing a quiz with the teachable agent.	231
7.3.4	The virtual beer-garden environment of the social game <i>SOAP</i> in the <i>AAA</i> engine.	232
7.3.5	The interaction scenario of the <i>SOCIALCOACH</i> application in different settings.	234
7.3.6	Some of the virtual characters that the user meets during the job application training. . .	234
7.3.7	The interaction scenario of the <i>ROBOTPUZZLE</i> application on the shared workspace.	235
7.3.8	Robot and user in the <i>ROBOTPUZZLE</i> are constantly following each other's attention. . . .	236
7.3.9	Robot and user in the <i>ROBOTPUZZLE</i> are disambiguating verbal reference with gaze.	236
7.3.10	Applications in which the <i>Robopec</i> robot <i>Reeti</i> was used as game-playing companion. . . .	237
7.3.11	Applications which use the <i>RoboKind</i> robots <i>Zeno</i> and <i>Alice</i> empathic partners.	238
7.3.12	An application in which the <i>KRISTINA</i> agent interacts with elderly and migrants.	238

7.3.13 Some photos taken during field tests in the context of nationwide promotion programs. . 239

LIST OF LISTINGS

5.3.1	The <i>BFQL</i> encoding of the feature structure from Figure 5.3.1 as list-based <i>PROLOG</i> term.	121
5.3.2	The <i>SWI-PROLOG</i> implementation of the basic feature structure predicate <i>val/3</i> .	122
5.3.3	The <i>SWI-PROLOG</i> implementation of the basic <i>BFQL</i> predicates <i>del/1</i> , <i>add/1</i> and <i>rll/2</i> .	125
5.3.4	The <i>SWI-PROLOG</i> implementation of the <i>BFQL</i> garbage collection predicate <i>clean/2</i> .	125
5.3.5	The <i>SWI-PROLOG</i> implementation of the basic predicates <i>mode/2</i> and <i>eqmode/2</i> .	126
5.3.6	The <i>SWI-PROLOG</i> implementation of the commonly used timeout predicate <i>timeout/2</i> .	126
5.3.7	The <i>SWI-PROLOG</i> implementation of the signal predicates <i>signal/2</i> and <i>detect/2</i> .	127
5.3.8	The <i>SWI-PROLOG</i> implementation of the temporal relation predicate <i>during/2</i> .	127
5.3.9	The <i>SWI-PROLOG</i> implementation of the ordering relation predicate <i>oldest/3</i> .	129
5.3.10	The <i>SWI-PROLOG</i> implementation of the <i>BFQL</i> quantifiers <i>formost/3</i> and <i>forfraction/4</i> .	130
6.2.1	The <i>SWI-PROLOG</i> implementation of the disambiguation predicate <i>disambiguate/2</i> .	163
7.1.1	The redefined production rules for command types in the <i>EBNF</i> grammar of the <i>SFGL</i> .	206
7.1.2	The redefined production rules for definition types in the <i>EBNF</i> grammar of the <i>SFGL</i> .	206
7.1.3	The redefined production rules for type definitions in the <i>EBNF</i> grammar of the <i>SFGL</i> .	206
7.1.4	The new production rules for variable definitions in the <i>EBNF</i> grammar of the <i>SFGL</i> .	207
7.1.5	The new production rules for function definitions in the <i>EBNF</i> grammar of the <i>SFGL</i> .	207
7.1.6	The new production rules for class path definitions in the <i>EBNF</i> grammar of the <i>SFGL</i> .	208
7.1.7	The new production rules for variable assignments in the <i>EBNF</i> grammar of the <i>SFGL</i> .	208
7.1.8	The new production rules for method invocations in the <i>EBNF</i> grammar of the <i>SFGL</i> .	208
7.1.9	The redefined production rules for expressions in the <i>EBNF</i> grammar of the <i>SFGL</i> .	209
7.1.10	The new production rules for variable expressions in the <i>EBNF</i> grammar of the <i>SFGL</i> .	209
7.1.11	The new production rules for invocation expressions in the <i>EBNF</i> grammar of the <i>SFGL</i> .	210
7.1.12	The new production rules for records and literals in the <i>EBNF</i> grammar of the <i>SFGL</i> .	210
7.1.13	The new production rules for scene scripts in the <i>EBNF</i> grammar of the <i>SFSL</i> .	211
7.1.14	The new production rules for scene definitions in the <i>EBNF</i> grammar of the <i>SFSL</i> .	212
7.1.15	The new production rules for turn definitions in the <i>EBNF</i> grammar of the <i>SFSL</i> .	212
7.1.16	The new production rules for utterance definitions in the <i>EBNF</i> grammar of the <i>SFSL</i> .	213
7.1.17	The new production rules for utterance elements in the <i>EBNF</i> grammar of the <i>SFSL</i> .	213
7.1.18	The new production rules for utterance actions in the <i>EBNF</i> grammar of the <i>SFSL</i> .	213
7.1.19	The new production rules for action features in the <i>EBNF</i> grammar of the <i>SFSL</i> .	214
7.2.1	The execution of a scene using the <i>playScene</i> function of the class <i>RuntimePlayer</i> .	223
7.2.2	The execution of a query using the <i>query</i> member function of the class <i>JPLEngine</i> .	225
7.2.3	The execution of a query using the <i>query</i> member function of the class <i>Evaluator</i> .	226

PART I

INTRODUCTION AND BACKGROUND

*“Acting is all about timing—
I mean, who has better timing than the MCs?”*

ARTIS LEON IVEY JR., A.K.A. COOLIO

CHAPTER 1

INTRODUCTION — JOINT ACTIVITIES WITH SOCIAL COMPANIONS

You can think whatever you like about the acting skills of Coolio, but his statement about the importance of timing for an acting performance is profoundly true. The timing of the actors' actions and utterances and the mutual coordination with their acting partners are especially important when they do not stick to a dictated storyline. In this case, they cannot cling to a script which prescribes the temporal sequence and meshing of the actors' lines and actions as well as the exact time and duration of pauses or the perfect co-verbal alignment of non-verbal behaviors, such as gestures, postures and facial expressions, down to the last detail. Instead, they need to improvise and spontaneously master the timing, synchronization, and interlocking of their behaviors during the course of the interaction by managing the reciprocal and dynamic process of unfolding interpersonal coordination of their behaviors. They mutually ground their knowledge, beliefs, assumptions, intentions, and ideas of the evolving plot with the aim to create a coherent and, in the best case, enthralling dramaturgy.

*THE ACTING
METAPHOR*

Very similar phenomena of mutual coordination and adaptation can be observed when watching a jazz orchestra playing a piece of music. Even with the sound turned off, one can witness various forms of synchrony, swing, and coordination while the band members skillfully improvise on their instruments. The musicians' breathing patterns, facial expressions, gestures, and the sways of their bodies are locked to the rhythm of the music. The rhythm of the improvisation is almost magically adopted by all musicians and determines the tempo and style in which they play together. Individual notes from the different instruments occur precisely at the same instant of time and notes from the individual musicians tend to begin and end simultaneously. The musicians constantly monitor and listen to their partners' play and seamlessly adapt their own play and timing by spontaneously re-planning their improvisation in response. As a result, the music played by each instrument and musician complements and intertwines with the others such that they together make the groove of the song.

*THE MUSIC
METAPHOR*

Those that are not gifted with a talent for acting or music can experience similar phenomena in team sports. For example, when watching a well-rehearsed, perfectly harmonized tennis doubles team, we can witness a smooth meshing of simultaneous or non-randomly patterned,

*THE SPORTS
METAPHOR*

well-timed, rhythmic activities and perfectly matched behaviors of the two doubles partners. Besides spontaneously reacting to environmental changes, such as wind gusts or odd spots on the court during a rally, they constantly monitor their opponents' and partners' actions and movements and adapt their own play accordingly. They are simultaneously advancing to the net after an aggressive and powerful serve or are rhythmically moving sideways at well-aligned staggered intervals in order to reduce angles and close gaps. They observe each other's movements and nonverbal actions to adapt their tempo and rhythm by synchronizing their split steps with their opponents' counter-movements and partners' ball contacts.

1.1 General Motivation

COORDINATION & GROUNDING MECHANISMS The reciprocal, dynamic, and incremental processes of mutual coordination and grounding, that can be observed in acting, music, and sports, are underlying nearly all natural interactions. They are especially evident in everyday *social* and *collaborative joint activities*, for example, when people are chatting with their friends in a bar, assembling a puzzle together with their children, or watching family or holiday photos with their grandparents. Humans consciously and unconsciously use the appropriate behavioral patterns and mannerisms to mutually coordinate on the content, process, and progress of such social interactions and naturally master this task already from birth on, without being aware of it most of the time. Even if the individual underlying behavioral mechanisms are difficult to identify and can hardly be systematically delineated from each other, it should be obvious that they exist. A closer look at the social and behavioral sciences as well as interdisciplinary fields helps to theoretically ground and understand these interactional phenomena.

TOWARDS SOCIAL COMPANIONS Many scientist showed that there exists a close link between the interaction partners' ability to coordinate, and especially synchronize, with their partners' behaviors and the perceived quality of their interaction in terms of social bonding effects, such as rapport building and affection development, the smoothness of their social encounter, and the efficiency of their collaboration. These findings from social science and behavioral psychology bear very promising perspectives for the research in building social user interfaces, in particular, socially competent and artificially intelligent companion technologies, such as *social humanoid robots* and *embodied conversational agents*. However, the interpersonal mechanisms of coordination and grounding that seem to work so well and intuitively in natural human interactions already from birth on, still work by no means as perfectly in human-agent interactions.

THE GENERAL MOTIVATION OF THIS THESIS The general motivation of this dissertation is, first, to pin down and deeply understand these interpersonal coordination and grounding phenomena and identify their underlying behavioral mechanisms, and, second, to design and implement a practicable and expressive modeling framework that allows the simulation and coordination of these functions in a social agent's computational behavior and interaction model. To avoid losing the focus, this thesis concentrates on investigating and simulating the roles of different gaze behaviors, speech overlaps and turn interruptions in physically situated and collaborative joint activities during dyadic interactions between a human user and a social agent.

1.1.1 Joint Activities

Whether musicians in jazz orchestras, players in tennis doubles, actors in improvisational theater, or participants of social interactions in everyday life, humans frequently engage in collaborative *joint activities* (Clark, 2005; Sebanz *et al.*, 2006; Huang *et al.*, 2015). The work in this dissertation is centered around such dynamic and physically situated interactions (Bohus and Horvitz, 2010b) while other interaction types, such as conversations via telephone or email exchange, are out of its scope. These interactions exhibit various aspects of rhythm, timing, synchronization, meshing, and coordination of interpersonal behaviors (Bernieri and Rosenthal, 1991). They are interactive tasks in which the involved parties are working together to coordinate attention, intention, knowledge, beliefs, assumptions, communication, and collective actions to achieve a common goal (Clark, 1996; Sebanz *et al.*, 2006).

*SITUATED &
SOCIAL JOINT
ACTIVITIES*

The initially described interactional phenomena observed in joint activities are most prominently known as *interpersonal coordination* (Bernieri and Rosenthal, 1991; Richardson *et al.*, 2005; Schmidt *et al.*, 2012) and *grounding* (Clark and Wilkes-Gibbs, 1986; Clark, 1996, 2005). They can be considered as a biological heritage (Crown *et al.*, 2002) since humans master some aspects of interpersonal coordination and grounding already from birth and are learning others easily throughout their life. Interpersonal coordination and grounding take place by means of linguistic and nonverbal behavioral signals, such as gaze cues, gestures, postures and facial expressions but also by material signals which are actions in which people deploy material objects, locations, or actions around them (Clark, 2005).

*INTERPERSONAL
COORDINATION
& GROUNDING*

Gaze has major social and regulatory functions for interpersonal coordination and grounding. It is involved in the production and recognition of turn-taking signals (Kendon, 1967; Duncan, 1972), footing and regulating the participant roles (Nielsen, 1962; Duncan and Fiske, 1977), and eliciting back-channel cues (Kendon, 1967; Allwood *et al.*, 1993; Bavelas *et al.*, 2002). It is used to direct and follow the partners' attention, predict their intentions (Sebanz and Knoblich, 2009; Baron-Cohen *et al.*, 2001; Meltzoff and Brooks, 2001), and is involved in the multi-modal disambiguation of references (Meyer *et al.*, 1998; Hanna and Brennan, 2007; Richardson *et al.*, 2007a). Finally, it plays a role in the reaction to cognitive and emotional displays (Kendon, 1967; Argyle and Cook, 1976; Doherty-Sneddon and Phelps, 2005), and interpersonal intimacy regulation (Kendon, 1967; Argyle and Dean, 1965; Abele, 1986).

*THE DIFFERENT
FUNCTIONS OF
GAZE BEHAVIOR*

Voice overlaps can serve as cooperative feedback to signal understanding, agreement, interest, engagement, and co-participation (Yngve, 1970; Allwood *et al.*, 1993) or enthusiastic listenership and high involvement (Tannen, 1984), but also a lack of interest, indifference, impatience, and non-support (Zimmerman and West, 1975). Interruption attempts signal the intention to change the topic or takeover the other's right to speak (Bennett, 1981; Schegloff, 2000; Tannen, 1994), for example, with the aim to grab the floor and dominate the conversation (Karakowsky *et al.*, 2004; Youngquist, 2009) or to reduce the partners' role as communicators (Kennedy and Camden, 1983b,a; Smith-Lovin and Brody, 1989). However, they can also show active mediation and refinement of thoughts (Shriberg *et al.*, 2001) or be a sign for activity and domain expertise in collaborative problem solving (Oviatt *et al.*, 2015).

*THE FUNCTIONS
OF OVERLAPS &
INTERRUPTIONS*

1.1.2 Social Companions

THE VISION
OF SOCIAL
COMPANIONS

There is a growing interest and effort in the creation of artificially intelligent and social agents that can naturally interact with humans. Visionaries from various arts and sciences are predicting a future in which they play important roles in our social environments and have great influence on the way we live our daily lives. For example, forward-thinking industrial managers are expecting them to support us in manufacturing processes. Stakeholders in health- and elderly care can hardly wait to deploy them for assisting nurses in hospitals and as caregivers for elderly people in retirement homes. Psychologists and pedagogues are placing great hopes on the possibilities to use them as educational peers for children and for the therapy of autistic people. Human resources managers are flirting with the idea of virtual recruiters that make their candidate selection processes more effective and cost-saving. A similar motivation lets coaching agencies hope to use them as trainers and mentors in safe and controlled virtual training environments. Others are seeing them as informational agents in public places or as personal companions in the comfort of our homes. There, they are expected to serve as lifestyle advisors in health-, fitness, or daily leisure activity management.

THE RECENT
DEVELOPMENT
& PROGRESS

While some of these visions and applications will eventually remain dreams of the future, the insights from the social and communication sciences as well as the breathtaking pace of the technological progress appears to be bringing social experiences with intelligent machines closer within our grasp. During the last decades, scientists and engineers have made real progress in the development of useful artificially intelligent, virtual characters and embodied conversational agents (Cassell *et al.*, 2000b; Rist *et al.*, 2003; Pelachaud, 2005) as well as social and collaborative robots (Fong *et al.*, 2003; Leite *et al.*, 2013) that might eventually become an integral part of our everyday life. These, for example, serve us by playing the role of playmates (Gebhard *et al.*, 2008; Behrooz *et al.*, 2014), trainers and coaches (Damian *et al.*, 2015; Traum *et al.*, 2008) or assistants in health- (Stiehl *et al.*, 2006; Kenny *et al.*, 2007; Breazeal, 2011) and elderly care (Heerink *et al.*, 2008, 2009; Broekens *et al.*, 2009).

SOCIAL SKILLS
& COMPETENCE
DEVELOPMENT

All these use cases have in common that the therein used social agents must achieve a high level of social competence and behave according to social conventions and rules in order to fulfill their task, interact naturally, and be accepted by the human interaction partners. This, first and foremost, requires that they master the entire range of verbal and nonverbal behavioral aspects of interpersonal coordination and grounding that underlie and govern social interaction, and that humans almost automatically learn from the earliest age. Therefore, it is crucial for social companions to render account of the aforementioned roles of gaze cues, voice overlaps, and interruptions that are involved in the manifold behavioral mannerisms and patterns which have that important functions for interpersonal coordination and grounding. This becomes especially important when they are performing collaborative joint activities on shared workspaces in which they also interact via objects in the physical environment. The following Section 1.2 introduces an exemplary scenario of such a physically situated, collaborative joint activity between a human and a robotic social companion, based on which it illustrates the various behavioral skills that must be mastered by a social agent.

1.2 Introductory Scenario

Research in social robotics and artificially intelligent agents leans on the future vision that humanoid robots and virtual characters might eventually become social companions in our everyday life. The relevant research issues that are investigated in this thesis are motivated on the basis of a particular use case of such a robotic companion in a domestic setting. This application representatively captures a great many aspects of this vision and is therefore well suited to provide a memorable context for this thesis. It serves to illustrate the various functions of gaze cues, voice overlaps, and interruptions for interpersonal coordination and grounding and, in this, comprehensively clarifies why social companions in such applications must master these behavioral aspects in social interactions with human partners.

*A MEMORABLE
CONTEXT FOR
THIS THESIS*

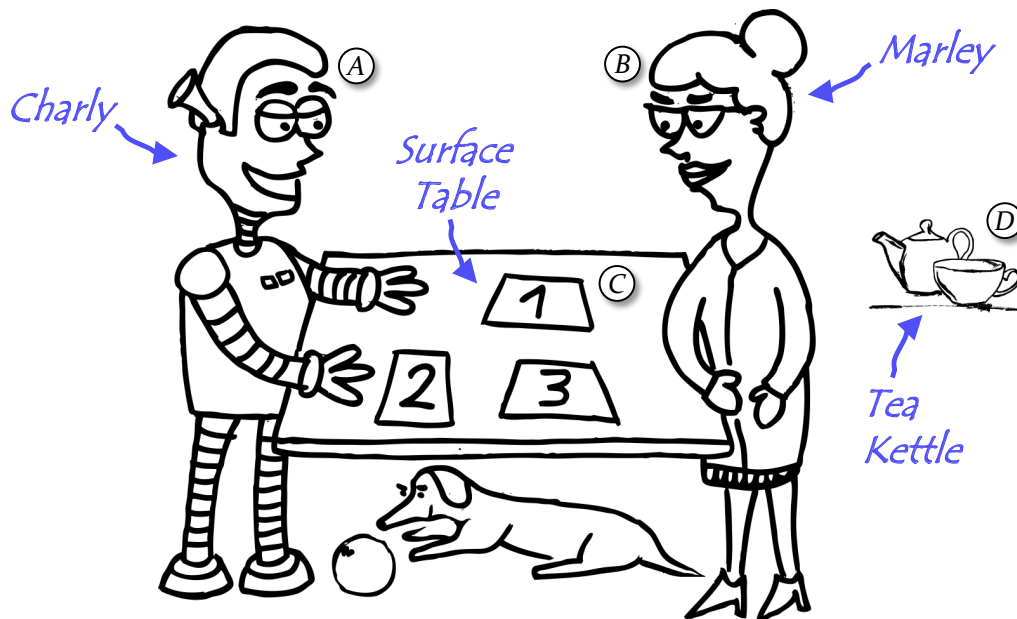


Figure 1.2.1: The physically situated, collaborative joint activity between Charly (A) and Marley (B).

Regarding the demographic growth in our society, it is quite conceivable that, in the near future, social robots will serve us as care-givers and service assistants in elderly care. Therefore, the illustrative scenario, shown in Figure 1.2.1, is from this domain and describes a joint activity between the personal robotic companion Charly (Figure 1.2.1 A) and Marley (Figure 1.2.1 B), an older woman who is an inhabitant of an assisted living project. One day, Marley invites Charly to drink a tea together while watching her family's holiday photos on a digital surface table in the living room (Figure 1.2.1 C). After putting the kettle on for a cup of tea (Figure 1.2.1 D), Charly starts the photo book application and drags some photos from the family's photo book onto the screen. Charly and Marley are positioned opposite to each other looking at the photos displayed on the shared workspace of the table. Marley is wearing glasses with an integrated eye-tracking system providing Charly with her eye gaze and field of view information in real-time. This enables Charly to keep track of the photos that Marley is looking at by using computer vision and object recognition algorithms.

*AN EXEMPLARY
COLLABORATIVE
JOINT ACTIVITY*

1.2.1 Interaction Extract

The following scenes represent a short extract of the interaction between Marley and Charly.

ATTENTION FOLLOWING & SIGNALING ① Charly is following Marley's gaze wandering across the table to those photos catching her attention. From time to time, Marley's gaze is lingering upon a specific photo for a slightly longer period of time.

INTENTION & INTEREST PREDICTION ② When Marley is particularly attracted by a photo, then he offers or provides information about it, asking, e.g. 'Shall I tell you about that?', 'Are you interested in this?' or saying 'This was in France!'

MULTI-MODAL REFERENCE DISAMBIGUATION ③ Suddenly, Marley is asking 'Tell me! Where is this beach?' while looking at one of the shown photos. Then she looks at Charly who returns the gaze and says 'This was your trip through France in 1980!'

ATTENTION DIRECTION & ALIGNMENT ④ Charly looks at another photo asking 'Where was that?' and observes Marley looking for this photo. Marley answers: 'That was our favorite restaurant ...' and meanwhile Charly utters some 'uh-huh"s.

COGNITIVE & EMOTIONAL STATUS DISPLAY ⑤ Marley pauses, slowly leans her head back and starts thoughtfully looking upwards for a moment. Charly attentively looks at her, says 'mm' and then raises his eyebrows and silently turns his head aside.

MIRRORING & INTIMACY REGULATION ⑥ Then, Marley looks directly at Charly with a big smile saying '!... we often enjoyed the sunset there.!' Charly immediately returns Marley's gaze and smiles back for a moment before looking away from her.

ENVIRONMENT & DISTRACTION MONITORING ⑦ Suddenly, the kettle starts ringing. Charly reflexively and sprightly looks at the kettle and Marley immediately follows Charly's gaze shift while Charly starts asking 'Marley, should I get a cup ...!'

BARGE-INS & TURN-CONFLICT REGULATION ⑧ But Marley promptly looks back to the photo and in turn interrupts Charly with 'I miss these old times.!' Thereupon, Charly immediately stops speaking and directly follows Marley's attention to the photo again.

TURN-TAKING SIGNAL PRODUCTION ⑨ After both kept staring at the photo for a moment, Marley asks 'Get what?' and then looks at Charly. Charly returns the gaze and rephrases 'Would you like some tea?' whereupon Marley replies 'No, thanks!'

FEEDBACK ELICITING & BACK-CHANNELS ⑩ Marley looks aside and starts saying 'You know —... and then looks in Charly's face who looks back, then lifts his eyebrows and nods a few times before Marley looks away again and says 'I'll take a nap!'

1.2.2 Scenario Discussion

In the above scenes, Charly constantly follows Marley's gaze to the photos in order to signal attention, interest, and engagement (①). By sharing her perceptual ground, he hopes to recognize her interest in specific photos and to predict her intention to receive more information about them (②). This also helps him to consider her gaze direction for the disambiguation of her verbal references to photos (③) and, thus, to avoid lengthy clarification dialogs. In turn, he uses his own gaze to direct Marley's attention to certain photos that he wishes more information about (④). He uses gaze cues to reveal his own thoughts and emotions, and interprets her gaze behavior to assess her mental state like comprehension problems or speech planning delays (⑤). He imitates her emotional displays whenever she looks at him with a facial expression to signal empathy and create rapport. However, he avoids staring at her and occasionally averts gaze to balance the interpersonal intimacy (⑥). He also monitors the environment and shifts his attention to unforeseen distractions (⑦). He politely ensures a smooth conversational flow by leaving the floor to Marley in case of a speech overlap that might be an interruption attempt (⑧). He aims for seamless and conflict-free turn exchanges by waiting for clear signals in form of mutual gaze and sufficiently long speech pauses until taking the floor (⑨). Finally, he constantly provides back-channel signals whenever Marley is eliciting such feedbacks by looking at him during her utterances and actions (⑩).

*THE DIVERSITY
OF BEHAVIORAL
FUNCTIONS*

The short episode might only take a minute or two but representatively illustrates that the participants of physically situated, social joint activities exploit versatile voice and gaze behaviors that essentially contribute with their complex interplay to interpersonal coordination and grounding. The use case can certainly be transferred to similar joint activities, such as collaborative assembling tasks on a shared workspace or many of the social interactions introduced at the very beginning of this thesis. It might also be generalizable to multi-agent and multi-user interactions in which very similar behaviors as in the dyadic case can be found.

The example also clearly illustrates that the proper coordination of the versatile social and regulatory functions of voice and gaze for interpersonal coordination and grounding is of prime importance. This coordination is the fundamental prerequisite for creating the impression of plausible and intelligent social behavior that supports intuitive social interaction experiences and promotes the users' willingness and interest to further interact with the agent. Allowing small discrepancies in the timing, synchronization, and interleaving or in handling priorities of behavioral functions and the underlying behavioral processes might let the agent appear unnatural, unbelievable, clumsy, simple-minded, or awkward. Disturbing or confusing the interplay of behavioral functions can thus very rapidly destroy the impression of an intuitive and smooth interaction. Such disturbances can manifest themselves in a variety of ways, such as excessively long periods of unintended silence or speaking overlaps, a lack of responsiveness and missing attention, or simply the impression of total incompetence and ignorance. Even if the user might not be able to exactly identify the specific reasons for an annoying perception, the interaction experience as a whole can be perceived as awkward and may distract the interaction partners from their common goal of the social joint activity.

*COORDINATION
OF BEHAVIORAL
FUNCTIONS*

1.3 Research Objectives

CREDIBLE
& NATURAL
BEHAVIOR

As a computer scientist, I want to enable myself and others to build social robots and virtual characters like Charly from the introductory example in Section 1.2. These social companions must behave in a human-like manner, that means they need to show a natural and plausible behavior, and have to support an intuitive, smooth, pleasant, consistent, and engaging interaction in order to be accepted by their human interaction partners (Loyall, 1997). This involves expertise in many disciplines and comes along with a variety of challenges (Vinayagamoorthy *et al.*, 2006). Among those are the development of technologies for expressive speech synthesis (Schröder, 2008) and natural language understanding (Gratch *et al.*, 2002) as well as the animation of facial expressions, gestures, and postures (Kipp *et al.*, 2007) and their recognition and interpretation (Wagner *et al.*, 2013; André *et al.*, 2014). Furthermore, it requires user models of emotion and personality (Gebhard, 2005; Gratch *et al.*, 2009) as well as suitable methods for dialog management (Traum *et al.*, 2008; Gebhard *et al.*, 2012; Ultes and Minker, 2014).

BEHAVIOR &
INTERACTION
MODELING

These technologies must be reasonably brought together in an agent’s computational *behavior and interaction model* which is responsible for the proper coordination of the numerous aspects of interactive behavior (Gebhard *et al.*, 2012; Mehlmann *et al.*, 2016). The vision described in the very beginning of this thesis and the illustrative scenario in Section 1.2 have shown that among the greatest challenges for such a social agent’s behavior and interaction model is the close coordination, that means the proper synchronization and prioritization of the many behavioral functions, or in better words, their underlying simultaneous, reciprocal, incremental, and highly interwoven behavioral and computational processes that contribute to interpersonal coordination and grounding with their complex interplay.

MODELING
CHALLENGE,
SUBTASKS &
REQUIREMENTS

The overall research goal of this thesis is the design of a modeling approach that allows building such behavior and interaction models for social agents. To this end, this thesis divides this challenge, the *modeling task*, into several attainable subgoals, called *modeling subtasks*, that an author must accomplish to create a well-engineered behavior and interaction model. These are further refined into clearly defined, task-specific subgoals, called *modeling requirements*, that the modeling approach for each subtask must fulfill to enable the author to master this task. Each of them is tackled with a number of *modeling concepts* that together form a *modeling language* for a specific subtask. These are finally combined to a *modeling language ensemble* which constitutes the *modeling framework* in this thesis.

1.3.1 Coordinating Functions and Processes

The first goal is simulating the complex interplay of the highly interwoven behavioral functions contributing to interpersonal coordination and grounding in human interactions. This requires modeling the *incremental and reciprocal meshing* of numerous, parallel and nested, behavioral and computational processes on different behavioral levels. The modeling approach proposed in this thesis tackles this requirement with a specially designed *state-chart variant* (Harel, 1987; Harel and Politi, 1998), called *Behavior Flow State-Charts (BFSCs)*. The

parallel and hierarchical structuring of a model is achieved through the *parallel decomposition* and *hierarchical refinement* of *BFSCs*. The quickly changing prioritization and seamless transitions between behavioral functions as well as their consistent reconstruction after suspensions require an adequate mechanism for the immediate *interruption and coherent resumption* of the underlying processes. For that purpose, *BFSCs* implement special *interruption policies* and an exhaustive *interaction history* mechanism.

1.3.2 Integrating Input and Context Events

The second goal is the robust understanding of the interaction partners' behaviors in human interactions. This requires the integration of information distributed over multiple modalities and context knowledge. The according input events can occur irregularly, have varying processing delays, and carry heterogeneous data from different processing stages. Therefore, they must be represented with a *uniform representation format* and maintained in a *well-organized working memory* to preserve their actual chronological order. Therefore, the modeling approach proposed in this thesis uses *feature structures* (Kasper and Rounds, 1986; Carpenter, 1992) that are managed in a logic *PROLOG fact base* (Clocksin and Mellish, 1981). It uses an embedded, domain-specific, *logic calculus*, called *Behavior Flow Query Language (BFQL)* for the *multi-modal fusion and reasoning*. This comprises logic predicates to evaluate semantic, temporal and quantitative integration constraints between multi-modal event as well as dynamic *garbage collection* predicates to manage the *event history*.

1.3.3 Creating Behavior and Dialog Content

The last goal is to create *versatile compositions of behavior* that resemble the wide range of behavioral and linguistic repertoire and natural variations of human-like behavior in social interactions and joint activities. Therefore, the modeling approach proposed in this thesis allows defining different types of *behavioral activities* using the *Behavior Flow Script Language (BFSL)*. These can be used to specify behaviors ranging from individual actions and non-verbal cues, such as gestures, postures, facial expressions, head, eye, and gaze movements, over verbal contributions and multi-modal utterances, to whole interactive performances of multiple agents. To create competent and informed behavior and dialog content, behavioral activities support the *flexible integration of knowledge* using the inline insertion of values or substitution of placeholder variables. Finally, the grouping and blacklisting of behavioral activities allows the *automatic variability of behavior* which avoids repetitions that would have a negative impact on the plausibility and naturalness of an agent's behavior.

1.4 Thesis Organization

The organization of this thesis is rather straightforward, in the sense that the applied scientific approach and research methods automatically prescribed the structuring of the thesis. The document is organized into four parts that are altogether consisting of eight chapters which reflect the individual milestones of the scientific approach depicted in Figure 1.4.1.

1.4.1 Scientific Approach

SCIENTIFIC
APPROACH &
MILESTONES

Figure 1.4.1 shows an illustration of the scientific approach with the individual research and development tasks that had to be carried out on the road to meeting the research objectives. Based on the motivating scenario (Figure 1.4.1 A), I started with a literature survey in social psychology and behavioral sciences to get a better understanding of interpersonal coordination and grounding and the underlying functions of gaze behavior, speech overlaps, and interruptions (Figure 1.4.1 B). Afterwards, I systematically identified and investigated the modeling challenges, tasks, and requirements that a modeling approach is faced with when simulating these functions and their interplay in an agent's behavior and interaction model (Figure 1.4.1 C). A review of related work provided an overview of state-of-the-art research on interpersonal coordination and grounding in human-agent interaction and modeling approaches for multi-modal fusion, interaction and dialog modeling, and behavior specification (Figure 1.4.1 D). The subsequent design of the modeling approach draws on valuable ideas from this work but goes beyond it by being the first to combine the advantages of hierarchical and concurrent state-charts, logic programming and template-based behavior descriptions in a novel modeling framework (Figure 1.4.1 E). This framework was afterwards applied in the development of various behavior and interaction models to examine its suitability in terms of practicability and expressiveness (Figure 1.4.1 F). Its following realization included the redefinition and extension of the modeling language ensemble as well as the refactoring of the existing *VSM*³ authoring framework. The reengineered tool was validated in a number of applications in the context of field tests, research, and teaching projects (Figure 1.4.1 G). Finally, the conceptual and technical contributions of the thesis were reflected and future research directives identified (Figure 1.4.1 H).

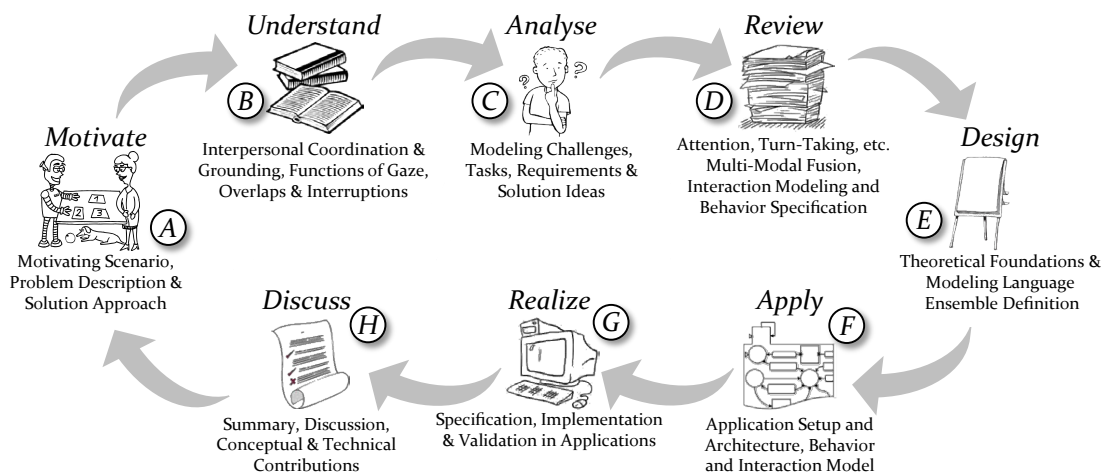


Figure 1.4.1: The overview of the scientific approach and the structuring of this dissertation.

1.4.2 Thesis Structuring

In Chapter 1, I explain my motivation for this work, place it in the scientific context, and highlight its relevance. I introduce an illustrative human-agent interaction scenario which serves as context throughout this thesis. Then, I describe my research objectives, identify

the challenges faced when realizing them, and outline the solutions that I designed to tackle them. Finally, I give an overview of the scientific approach and organization of this thesis.

In Chapter 2, I introduce the required terminology and background knowledge to provide a profound understanding of the interactional phenomena investigated in this thesis. Therefore, I review relevant literature from social and behavioral sciences and, in places, human-agent interaction. This survey comprehensively explains how the functions of gaze behavior, speech overlaps and interruptions contribute to interpersonal coordination and grounding. CHAPTER 2 —
BACKGROUND

In Chapter 3, I present the modeling challenges, tasks, and requirements faced by the proposed behavior and interaction modeling approach. They are illustrated based on the introductory scenario to highlight their importance for interpersonal coordination and grounding. I also briefly mention the concepts and formalisms that the modeling approach proposed in this thesis uses to tackle them, as far as this is needed for a comparison to related work. CHAPTER 3 —
CHALLENGES

In Chapter 4, I review relevant related work on multi-modal fusion, behavior and interaction modeling and multi-modal behavior specification in human-agent interaction. I discuss advantages and drawbacks of some selected state-of-the-art solution approaches, investigate to which extent they address the identified modeling tasks and requirements, and explain how the modeling framework proposed in this thesis overcomes many of their insufficiencies. CHAPTER 4 —
RELATED WORK

In Chapter 5, I discuss some design issues that I particularly paid attention to during the design of the modeling approach. Then, I present the architecture of the modeling framework which is divided into an ensemble of modeling languages, each of which tackles an individual modeling task and its task-specific requirements. Afterwards, I present the definition of each individual ensemble member and explain how it tackles these requirements. CHAPTER 5 —
CONCEPTION

In Chapter 6, I illustrate the modeling approach based on a particular use case which is very similar to the scenario in Chapter 1. I systematically develop a reusable and adaptable behavior and interaction model for the agent in this application. The understanding of this model gives a good idea of the best practice to use the proposed modeling approach for the development of interactive applications with virtual characters and social robots. CHAPTER 6 —
ILLUSTRATION

In Chapter 7, I explain how the conceptual design has been realized in a reference implementation. I explain how lexical and syntactical extensions found their way into a reengineered specification of an existing modeling language ensemble. Then, I show how this ensemble has been implemented by refactoring the *VSM*³ authoring tool. Finally, I present some applications that have been developed with *VSM*³ to validate the modeling approach. CHAPTER 7 —
REALIZATION

In Chapter 8, I first present a short summary of the work presented in this thesis. Afterwards, I identify the conceptual and technical contributions of this thesis. In this, I also briefly revise the discussion of advantages and limitations of the proposed modeling approach and its technical realization. Finally, I give a view on potential extensions and further development possibilities of the approach and interesting future research directions. CHAPTER 8 —
CONCLUSION

BACKGROUND — INTERPERSONAL COORDINATION AND GROUNDING

The research objective of this thesis is the development of an *expressive* but nevertheless *practicable* modeling approach for the interactive behavior of social agents, such as virtual characters and embodied conversational agents (Cassell *et al.*, 2000b; Pelachaud, 2005; Rist *et al.*, 2003) or social robots (Fong *et al.*, 2003; Leite *et al.*, 2013). The major challenge is therein the design of particularly suited modeling formalisms that facilitate the proper coordination of the manifold behavioral aspects that contribute to *interpersonal coordination* (Bernieri and Rosenthal, 1991; Schmidt and Richardson, 2008; Lumsden *et al.*, 2012) and *grounding* (Clark and Brennan, 1991; Brennan, 1998; Clark, 2005). As described at the very beginning of this thesis, these two related phenomena underlie nearly all everyday social interactions and joint activities and are therefore also crucial for social agents to interact naturally and credibly.

The importance for social agents to master the behavioral functions contributing to interpersonal coordination and grounding has been illustrated and motivated by the introductory example scenario in Section 1.2. In line with the focus of this thesis, the example focused on the social and regulatory functions of *gaze behavior* (Kendon, 1967; Argyle *et al.*, 1973; Kleinke, 1986; Srinivasan and Murphy, 2011; Mutlu *et al.*, 2012; Jokinen *et al.*, 2013; Mehlmann *et al.*, 2014b; Ruhland *et al.*, 2015) as well as *speech overlaps* and *turn interruptions* (Bennett, 1981; Drummond, 1989; Tannen, 1994; Olbertz-Siitonen, 2009; Tannen, 2012) and the roles that their different functions play for interpersonal coordination and grounding.

To convey a profound understanding of the two interactional phenomena, I now introduce the fundamental terminology and required theoretical background knowledge by reviewing the relevant basic literature from social and behavioral sciences and, in places, mention related work from human-agent interaction. First, I present the definitions and theoretical foundations of interpersonal coordination and grounding in Section 2.1. Subsequently, I give an overview of the various functions of gaze behaviors in Section 2.2 before I discuss the different effects of speech overlaps and interruptions in Section 2.3. Some of the roles of these behavioral aspects for interpersonal coordination and grounding are illustrated and discussed on the basis of the introductory example scenario described in Section 1.2.

2.1 Interpersonal Coordination and Grounding

INTERPERSONAL
COORDINATION

The participants of social interactions continuously coordinate and adapt their motor movements and behaviors in a reciprocal and dynamic process of constant rebalancing. This often subconscious mutual interplay produces a seamless meshing and smooth temporal synchronization of their behaviors and gives rise to simultaneous and rhythmic joint actions and the mutual entrainment of their interaction speeds and rhythms. These aspects of coordination serve a fluent and effective regulation of the interaction flow and the organization of turn-taking while additionally having social functions such as improving and signaling closeness, rapport, or empathy. In social and behavioral sciences this mutual accommodation, interlocking, and synchrony of the interaction partners' behaviors is known as *interpersonal coordination* (Bernieri and Rosenthal, 1991; Richardson *et al.*, 2005; Schmidt *et al.*, 2012).

COMMON
GROUND
& GROUNDING

In addition to this mutual coordination of interactional speed, rhythm, and synchrony as well as the tight meshing of their behaviors, the interaction partners are, at the same time, constantly busy collaborating on the coordination of their shared knowledge, beliefs, assumptions, and intentions about the content and process of the interaction. They establish, maintain, and repair this common understanding of their joint activity using a variety of multi-lateral and multi-modal acknowledgment and clarification mechanisms. In this, they are always endeavored to choose these means such that they achieve their common goal with the *least collaborative effort* (Clark and Wilkes-Gibbs, 1986; Clark and Schaefer, 1989; Clark and Brennan, 1991). This idea of shared information and mutual understanding of what the interaction partners are doing is referred to as *common ground* and the process of its constant updating and accumulation is known as *grounding* (Clark, 1996; Brennan, 1998; Clark, 2005).

While it is intuitively obvious that these processes of mutual coordination and grounding exist, it is hard to tell what they really are and how they actually work and interplay. Furthermore, it seems that they are not disjoint but rather overlapping concepts and cannot always be clearly delineated from each other. A closer look at fundamental research in social psychology and behavioral sciences helps to identify and better understand these two phenomena. This is clearly necessary to motivate and theoretically ground the work in this thesis and helps to gain an impression of the difficulties that we are faced with when integrating and simulating the interplay of the individual behavioral aspects of interpersonal coordination and grounding with a social companion's behavior and interaction model.

2.1.1 Interpersonal Coordination

DEFINITION OF
INTERPERSONAL
COORDINATION

Interpersonal coordination is a fascinating phenomenon that is present in nearly all aspects of our social life (Bernieri and Rosenthal, 1991; Richardson *et al.*, 2005; Schmidt *et al.*, 2012). Its conceptualization is based on the observation that the participants' behaviors in natural social interactions are often rhythmic, non-randomly patterned, and synchronized. People are often entrained by each other's interactional rhythm and mainly subliminally adjust the timing and shaping of their behaviors to one another. Consequently, their behaviors are frequently similar or identical in form, occur at roughly or exactly the same time, and are

seamlessly intermeshed with each other. Sometimes they even coincide almost perfectly in both, form and time (Bernieri and Rosenthal, 1991; Lakin, 2012). Based on these observations, Bernieri and Rosenthal (1991) rather descriptively define *interpersonal coordination* as

“the degree to which the behaviors in an interaction are non-random, patterned, or synchronized in both timing and form”

Interpersonal coordination means acting in synchrony and synchronized as well as similar or identical to each other. It appears in everyday interactions in which individuals reciprocally coordinate their motor movements with respect to the rhythmic behavior of others. Thus, it can be considered as the complement of *intrapersonal coordination* which is the coordination of a person’s individual body segments among one another (Ramenzoni *et al.*, 2011).

Interpersonal coordination is involved when reciprocally matching or meshing with one another and carrying each other away physically, emotionally, or intellectually. It can be overtly controlled through physical contact but also be unintentionally be performed during visual interactions (Richardson *et al.*, 2007b). Thus, it underlies all the social activities mentioned in Chapter 1, such as dancing, music, sports, or simply smalltalk, but, of course, is particularly evident during the illustrative example scenario from Section 1.2. In this, it increases in degree and stability with the difficulty of collaborative tasks that require two or more people to coordinate with each other to attain a common goal (Ramenzoni *et al.*, 2011).

*OCCURRENCE OF
INTERPERSONAL
COORDINATION*

Interpersonal coordination manifests itself in many multi-modal and multi-directional behavioral patterns and mannerisms with various social and regulatory functions. For example, it ensures the trouble-free alternation of contributions (Sacks *et al.*, 1974) by properly recognizing and producing turn-taking actions (Nielsen, 1962; Kendon, 1967; Duncan, 1972; Clark, 1996). It includes carefully listening and observing the partner to achieve a well-timed start of the own contributions (Shriberg *et al.*, 2001) at possible completion points (Sacks *et al.*, 1974). It drives the interaction flow and rhythm when coordinating back-channels, minimal responses, and other accompaniment signals with the partner’s behavioral cues in order to signal co-participation (Schegloff, 1968; Fishman, 1997), engagement, and encouragement (Kendon, 1967; Yngve, 1970; Allwood *et al.*, 1993; Bavelas *et al.*, 2002). It also comprises correctly matching the partners’ gestures, postures, facial expressions, and behavioral mannerisms at the proper time to create rapport and involvement (Chartrand and Lakin, 2013).

*EXPRESSION OF
INTERPERSONAL
COORDINATION*

Because interpersonal coordination of behavior has the two different facets of *similarity* and *timing*, it is formally divided into *behavior matching* and *interactional synchrony* (Condon and Ogston, 1966, 1967; Kendon, 1970) which is also referred to as *interpersonal* or *social synchrony* (Hove and Risen, 2009; Marsh *et al.*, 2009). Behavior matching usually refers to mimicry or imitation phenomena where interaction partners perform the same or similar movements, actions, or behaviors. Complementary, interactional synchrony refers to the interaction partners’ synchrony or mutual entrainment in which the movements and behaviors of the partners become properly organized and intertwined in time.

*TWO FACETS OF
INTERPERSONAL
COORDINATION*

Behavior Matching

DEFINITION OF
BEHAVIOR
MATCHING

Behavior matching describes the *degree of similarity* between the interaction partners' observable behavior (Bernieri and Rosenthal, 1991). Bernieri and Rosenthal (1991) consider it as similarity measure and define that *behavior matching*

“occurs when two or more people show similar body configurations”

The described similarity can be the expression of an unconscious *emotional contagion* or the result of an, either intentional or unconscious, *behavioral mimicry*, imitation or mirroring of the partner's behavior (LaFrance, 1982; Louwerse *et al.*, 2012; Chartrand and Lakin, 2013). In this, the term *emotional contagion* usually means “*catching another's emotion*” (Hess and Fischer, 2013) and actually taking on the affective state that matches the other's emotional display. In contrast, the labels *mimicry*, *imitation*, or *mirroring* usually exclusively refer to the replication of the expressive component of the partners' displayed behavior (Hess and Blairy, 2001; Hess and Fischer, 2013) without actually empathizing with the partner. In this, *mimicry* and *mirroring* usually relates to an automatic reaction, without conscious awareness, while *imitation* means an intentional acting. For example, *behavioral mimicry* has descriptively been defined by Chartrand and Lakin (2013) as

“*the automatic imitation of gestures, postures, mannerisms, and other motor movements*”

The tendency to unconsciously mimic postures, gestures, and facial expressions of interaction partners is also referred to as *chameleon effect* (Chartrand and Bargh, 1999; Lakin *et al.*, 2003). A very similar phenomenon which is sometimes also used interchangeably with mimicry is *mirroring* (LaFrance, 1982; Chartrand and Bargh, 1999), which means that two persons take poses that are the exact *mirror images* of each other.

FUNCTIONS OF
BEHAVIOR
MATCHING

Behavior matching phenomena and their functions pervade all behavioral modalities, such as gestures (Chartrand and Bargh, 1999; Lakin and Chartrand, 2003), body postures (Kendon, 1970; LaFrance, 1982), and facial expressions (Bavelas *et al.*, 1986; Meltzoff and Moore, 1983) as well as simple motor movements, such as foot shaking (Chartrand and Bargh, 1999) and many more (Chartrand and Lakin, 2013; Lakin, 2012). It is believed that mimicry or mirroring, whether performed intentionally or unconsciously, reflect the congruence of internal mental states and attitudes toward each other (Schefflen, 1964). Interaction partners are considered more coordinated to the extent that their mental states and external behaviors are more alike and matched. This suggest that these behaviors are among the fundamental behavioral repertoire to create social effects like trust, rapport, liking, emotional contagion, and empathy (van Baaren *et al.*, 2009; Chartrand and van Baaren, 2009; Chartrand and Lakin, 2013).

Much research has shown that interaction partners continually match one another's postures during an interaction (Condon and Ogston, 1966, 1967; Kendon, 1970; Shockley *et al.*, 2003)

and are thus judged to have a higher rapport with each other (Dabbs, 1969; Trout and Rosenfeld, 1980). Furthermore, listeners tend to mirror a speaker's posture whom they find engaging (LaFrance, 1979, 1982). An affiliation goal increases unconscious mimicry and, vice-versa, rapport and affiliation increase mimicry (Lakin and Chartrand, 2003). While real empathy requires mind reading and perspective taking (Baron-Cohen, 1997), the ideomotor mimicry of facial expressions is a primitive method to convey the impression of empathy (Bavelas *et al.*, 1986, 1987). Already neonates and infants imitate facial gestures (Meltzoff and Moore, 1983) and adults spontaneously mimic their interaction partners' facial expressions (McHugo *et al.*, 1985). In this, facial expressions are directly linked to the experience and communication of emotions to the interaction partner (Ekman, 1972, 1977; Ekman and Friesen, 2003). The facial mimicry of any emotional expression signals responsiveness and liking of the mimicker on the part of the mimickee and is a vehicle of empathy and social bonding (Kulesza *et al.*, 2015). In this, emotional mimicry is context-dependent in the sense that it functions as a social regulator taking into account the relationship between observer and expresser (Hess and Fischer, 2013). People mimic emotional signals not automatically but only when they believe these signals promote affiliation. Thus, people less likely mimic strangers and others they don't like even when the emotional display signals a negative emotion such as antagonism (Hess and Fischer, 2014). In turn, they, for example, perceive the partner as aversive and not willing to communicate if smiles are not timely answered (Cappella, 1997; Hess and Fischer, 2014; Gironzetti *et al.*, 2016).

Interactional Synchrony

Interactional synchrony is composed of three aspects whose interplay enables highly reciprocally and dynamically synchronized interactions. First, *rhythm* can be observed in the mutual alignment of the interaction partners' walking rhythms, postural sways, breathing patterns, eye movements, and other rhythmic or cyclic behavioral aspects (Richardson *et al.*, 2008). Second, *simultaneity* can be observed in coincident posture changes and gaze movements or immediate facial mimicry. Finally, *meshing* or intertwining of behavior can be observed in the smooth exchange of turn-taking actions or in well-timed back-channels.

ASPECTS OF
INTERACTIONAL
SYNCHRONY

The *interaction rhythm* determines the tempo and style of the interaction which is necessary for all ordered interactions (Davis, 1982). Some interactions have a more rapid and jerky nature whereas others occur in a slower and more fluid fashion (Bernieri and Rosenthal, 1991). It is presumed that the rhythm of an interaction between two or more interaction partners is linked to, as Bernieri and Rosenthal (1991) say

DEFINITION OF
INTERACTION
RHYTHM

“the degree of congruence between their behavioral cycles”

The interaction partners can be more or less “*in sync*” regarding an aspect of the interaction that exhibits rhythmic or cyclic characteristics, such as the expression of engagement through cyclically re-occurring periods with specific multi-modal behavioral patterns (Stern, 1974) or the rhythmic occurrence of conversational activity in the course of an interaction

or joint activity (Hayes and Cobb, 1982). Via constant informational coupling, for example, by monitoring each other, over time, the movements of the interacting individuals become mutually entrained, which leads to the emergence of stable behavioral patterns (Lumsden *et al.*, 2012). Even after an interruption of an interaction’s rhythm by an unplanned event, the same stable rhythm reemerges after a short time again (Hayes and Cobb, 1982). In this, one partner often has the function of a *time giver* while the others are entrained to the same behavioral cycle as the time giver. The stability of this time giver role may be determined by the social relationship between the interaction partners, such as dominance and status (Baron and Boudreau, 1987). This connection between social relationship and interactional rhythm explains why some persons more efficiently entrain their interaction partners while others are not able to dictate the rhythm of an interaction.

DEFINITION OF BEHAVIORAL SIMULTANEITY *Simultaneity* means the simultaneous or at least chronologically very close occurrence of the interaction partners’ behaviors or actions. Any interpersonal behavior has identifiable moments at which the partners are moving or acting simultaneously or very tightly aligned in time. Bernieri and Rosenthal (1991) understand *simultaneity* simply as

“the co-occurrence of two or more behaviors”

In this, behavior may be understood in the broadest sense, which means muscle movements, nonverbal cues such as gestures, body postures, facial expressions, and gaze behaviors as well as vocalizations, whole utterances, or even emotional and mental states (Bernieri and Rosenthal, 1991). The criterion of co-occurrence and the granularity of the considered time frames that define the simultaneity of behaviors vary depending on the specific behavior. For example, body postures may be considered simultaneous within a time frame of several seconds while the time frame for determining the simultaneity of facial expressions spans only a few hundredths of a second.

DEFINITION OF BEHAVIORAL MESHING The last fundamental feature of interactional synchrony is *behavioral meshing*, which Bernieri and Rosenthal (1991) describe as

“the unification of two potentially random, non-patterned behavioral elements into a meaningfully described “whole” or synchronous event”

Following this definition, behavioral meshing means the ability of smooth intertwining and close interlocking of the interaction partners’ behaviors that are thereby sometimes complementing each other to form combined, multi-directional, and multi-modal behavioral patterns. The partners may synchronize with each other, such that their behaviors mesh smoothly or their behaviors may interfere and conflict which has the effect that the interaction feels awkward or clumsy. This enhancing influence of behavioral meshing on the efficiency and perception of the interaction becomes obvious when observing the close intermeshing of the interaction partners’ behaviors for the production of turn-taking patterns

that they use for the smooth organization of the participant's roles (Nielsen, 1962; Kendon, 1967; Duncan, 1972, 1974; Goffman, 1979; Goodwin, 1980, 1981) and the efficient exchange of their conversational contributions (Bernieri and Rosenthal, 1991). It is believed that the coordination with the partner's movements and speech helps to anticipate more accurately the termination of their vocalizations and completion points of turns at transition-relevant places (Sacks *et al.*, 1974), which, in turn, improves the promptness with which the next speaker starts a response without cutting off the current speaker's utterance (Dittmann and Llewellyn, 1967).

Research exploring the effects of interpersonal synchrony for sociality has found a number of benefits (Lumsden *et al.*, 2012; Chartrand and Lakin, 2013; Tschacher *et al.*, 2014). For example, interpersonal synchrony facilitates person perception by enhancing memory for an interaction partner's utterances and facial appearance (Macrae *et al.*, 2008). It is able to blur self-other boundaries by creating the perception of self-other similarity and social identification (Miles *et al.*, 2010b; Paladino *et al.*, 2010). It furthermore enhances altruistic behavior and compassion in the sense of an empathic concern for the well-being of others (Valdesolo and DeSteno, 2011). It improves cooperation by strengthening group cohesion and social attachment among group members (Wiltermuth and Heath, 2009). It increases liking and rapport (Hove and Risen, 2009) and entails affect between the interaction partners (Tschacher *et al.*, 2014). Vice-versa, humans synchronize their behaviors to a greater degree when interacting with others that they like (Bernieri, 1988; Bernieri *et al.*, 1994; Cappella, 1997). For example, people moving rhythmically in-phase or anti-phase, performing their actions at equivalent or opposite points of the movement cycle, are usually rated to have a higher degree of rapport (Miles *et al.*, 2009). People that are trying to reduce the perceived social distance between each other show a higher degree of synchrony (Miles *et al.*, 2011) while those that experienced antipathy in the past synchronize less in subsequent interactions (Miles *et al.*, 2010a). All these results suggest that the degree of liking, rapport, and cooperation is reflected by the quality of interactional synchrony between the interlocutors and that it promotes sociality (Marsh *et al.*, 2009; Schmidt *et al.*, 2012; Tschacher *et al.*, 2014).

FUNCTIONS OF
INTERACTIONAL
SYNCHRONY

A concept which is similar to interactional synchrony is *coordinated interpersonal timing* (Crown, 1991; Feldstein *et al.*, 1993; Hane *et al.*, 2003), sometimes referred to as *interpersonal timing* (Crown *et al.*, 2002). It has mainly been studied in the area of infant research and less in the field of adult social behavior (Crown, 1991). This research shows that the temporal patterning of social interaction is a fundamental aspect of behavior and lies in our nature and biologic heritage (Capella, 1981; Jasnow *et al.*, 1988; Feldstein *et al.*, 1993; Crown *et al.*, 2002; Hane *et al.*, 2003). It has descriptively be defined by Crown *et al.* (2002) as

“an alteration in the temporal patterning of one speaker's visual and vocal behavior in response to that of the other speaker's behavior”

Thus, coordinated interpersonal timing explicitly concerns the temporal relationship between the temporal patterns of two interacting partners and explicitly refers to visual be-

haviors and gaze patterns and their alignment with speech. The better the temporal pattern of each partner in a dialog is predictable from that of the other, the better is the interpersonal synchrony or degree of coordination (Feldstein *et al.*, 1993). The coordinated interpersonal timing is positive when the temporal behaviors are positively correlated and negative when they are negatively correlated. Positive coordinated interpersonal timing has a reinforcing influence on interpersonal attraction (Crown, 1991). Finding just the right degree of coordinated interpersonal timing can have a positive influence on affect and involvement. An optimum degree of rhythmic coordination in social interaction makes the interaction more rewarding such that the interaction partners will like each other (Warner *et al.*, 1987). In general, coordinated timing in adult interactions is a dimension of the communication of mood, empathy, psychological differentiation, and perceived interpersonal similarity (Capella, 1981; Feldstein and Welkowitz, 1987; Rosenfeld, 1987).

2.1.2 Grounding in Joint Activity

DEFINITION OF COMMON GROUND Besides interpersonal coordination, another important process that underlies all human interaction, especially joint activities and collaborative settings, like in the illustrative scenario from Section 1.2, is *grounding* (Clark, 1996; Brennan, 1998; Clark, 2005). This is a reciprocal, collective, and accumulative process that can be considered as the incremental and multi-directional updating of the interaction partners' *common ground* (Clark and Brennan, 1991). The common ground itself is considered as the mutual belief that the interaction partners have understood each other well enough for the current purpose. It has rather descriptively been defined by Clark and Brennan (1991) as

“the mutual knowledge, mutual beliefs, and mutual assumptions”

that the participants of an interaction believe to share with each other. Interaction partners constantly establish, maintain, and repair their common ground. This includes monitoring their own and the others' behaviors and actions, and eventually producing appropriate interventions and repairing mechanisms if the common ground is, or threatens to be, disrupted. Such disruptions arise from wrong presuppositions and misunderstandings due to missing attention or whenever one participant presumes sensory, perceptive or cognitive abilities that the other cannot serve with. The shared information constituting the common ground is used to coordinate on the content of the dialog or task of the joint activity as well as the process and progress of what they are doing (Clark and Brennan, 1991). However, it should not be understood as a quantity that can directly or indirectly be measured (Koschmann and LeBaron, 2003) but should rather be imagined as a cooperatively accumulated and distributed form of mental abstraction (Clark, 1996).

Contribution Theory

A BILATERAL MODEL OF INTERACTION The theory of a common ground arises from a *bilateral* model of natural conversations and social joint activities. These are actually more than just simple alternating sequences of iso-

lated utterances or actions, but rather consist of closely coordinated collective acts of speakers and listeners. In this, speakers and actors monitor their own behaviors and those of their addressees, and take both into account as they proceed with their action or utterance to ensure that they are being attended to and understood by the partners. These, in turn, constantly keep the speaker informed about their current state of understanding, letting him know when he has succeeded, which means that the speaker's contribution is mutually accepted and flown into the common ground (Clark and Krych, 2004). Thus, contributing to a discourse or collaborative activity is a bilateral cooperation and information exchange that requires each partner to monitor himself and the others, and to use the thereby gained information in further speaking and acting (Clark and Krych, 2004). This bilateral model extends the more traditional *unilateral* models of language production (Bock and Levelt, 1994; Ferreira, 2000) and understanding (Tanenhaus and Trueswell, 1995; Frazier and Clifton, 1997) that solely consider self-monitoring and self-repair (Schegloff *et al.*, 1977; Levelt, 1983).

The bilateral perspective is achieved by enlarging the frame of an interaction's analysis from the single message unit, like an utterance or action, to the notion of a reciprocally and collectively produced *contribution* (Clark and Schaefer, 1987, 1989). According to this *contribution theory*, these always have a *presentation phase*, in which the speaker produces an utterance or action, followed by an *acceptance phase*, in which the addressed partner may explicitly acknowledge, modify, clarify, or implicitly accept it by continuing with the next relevant contribution. Contributions may be nested within other contributions as parts of *clarification subdialogs*. Because an utterance presented by one partner does not become a contribution until it has been accepted by the other, both speakers and addressees are mutually responsible for what is contributed to the common ground.

A THEORY OF
CONTRIBUTION

Grounding Mechanisms

According to the contribution theory, an utterance presented by a speaker or an action performed by an actor is not part of the interaction's common ground until it has been accepted or approved by the addressee (Clark and Wilkes-Gibbs, 1986; Clark and Schaefer, 1987, 1989). This acceptance happens through the systematic exchange of *acceptance evidence* during the grounding process (Clark and Schaefer, 1987, 1989). Depending on their common goals, where necessary, the participants of an interaction adjust the *grounding criterion* which determines the degree of evidence that is required to accept a contribution (Clark and Wilkes-Gibbs, 1986; Clark and Schaefer, 1989; Wilkes-Gibbs and Clark, 1992).

GROUNDING
EVIDENCE &
CRITERION

The exchange of the evidence of a contribution's acceptance can be provided by various methods on multiple *levels of grounding* (Clark, 1996; Clark and Krych, 2004). Examples for them can be found in the following exchange that could have been occurred as an alternative course or subdialog of the conversation between Charly and Marley starting after scene ③ of the introductory example scenario from Section 1.2:

LEVELS OF
GROUNDING

- ③ Suddenly, Marley is asking 'Tell me! Where is this beach?' while looking at one of the shown photos. Then she looks at Charly who returns the gaze and says 'This was your trip through France in 1980!'

- ④ Marley knits her eyebrows asking 'Through what?' — Charly repeats 'France!' and she says 'Aha!'.
- ⑤ Marley looks at Charly with wondering eyes asking 'The one in the Champagne or in Aquitaine?'. Charly answers 'The second trip!' and Marley then acknowledges that by a nod of her head.
- ⑥ After a moment she starts smiling and says 'Yes, we were eating tonnes of oysters each day!'.

GROUNDING BY ATTENTIVENESS

First, scene ③ illustrates that on the level of *attentiveness*, common ground is established by paying attention to each other which is usually achieved by constantly monitoring and carefully listening to each other. If an addressee misses to pay attention then he might have difficulties to acoustically understand the speaker in a conversation or to visually follow an actor's actions in a joint activity. In this case, he would try to re-establish the common ground and signal that he has been inattentive by saying "What?", "Sorry?" or "Again!" and the speaker would ordinarily repeat his utterance or action in response.

GROUNDING BY IDENTIFICATION

Second, on the levels of *identification* both interaction partners must ground their belief that the speaker's words, phrases, and sentences have been completely identified. An example of an identification grounding can be found in scene ④ of the alternative scenario above when Carly answers Marley's question about a photo saying "This was your trip through France in 1980!". Marley signals that she has correctly identified and understood the sentence with the exception that she is uncertain about the last word by asking back "Through what?". In order to repair the common ground, Charly then explicitly repeats the word by saying "France!" and she acknowledges the identification of the word by responding with "Aha!".

GROUNDING BY UNDERSTANDING

Then, on the level of *understanding* both interaction partners must ground their belief that the speaker's words, phrases, and sentences have been correctly understood and interpreted. Disruptions of the common ground on this level are usually repaired by initiating some kind of *clarification subdialog*. An example of this kind of grounding can be found in scene ⑤ of the alternative scenario above when Marley asks back "The one in the Champagne or in Aquitaine?" and Charly provides the clarification by saying "The second trip!", in this, relying on the fact that they both already share the knowledge that the second trip was the one to Aquitaine as part of their common ground. In response, Marley finally confirms understanding and agreement by producing a back-channel signal in form of a head nod.

GROUNDING BY CONSIDERATION

Finally, on the level of *consideration*, both interaction partners must be able to reasonably and coherently continue their conversation or joint activity based on the recently updated common ground. For that purpose, they produce reasonable and appropriate next contributions to the interaction by answering a question or continuing with a question on their own. An example of such a strategy can be found in scene ⑥ of the alternative scenario when Marley is continuing with the utterance "Yes, we were eating tonnes of oysters each day".

METHODS OF GROUNDING

The exemplary subdialog also illustrates that the interaction partners may generally employ different methods of accomplishing acceptance in a contribution and to ground their interaction at the aforementioned four levels (Clark, 1996; Clark and Krych, 2004). These methods mainly differ in the degree of evidence for grounding that they provide and the level

of grounding on which they may be useful and reasonable. For example, responding to an utterance by simply repeating a part or all of what has been said by the partner is a very strong and explicit type of grounding evidence which is provided on the levels of identification and understanding. Another possibility for providing positive evidence of the common ground, but a somewhat weaker explicit method, is the use of an acknowledgment or continuer, such as for instance “yes”, “uh”, “okay”, and others. More information is already provided by the use of assessments such as “gosh”, or “jipiiee”, or nonverbal social signals such as emotional facial expressions (Ekman and Friesen, 1969; Ekman, 1977). Listeners in conversations provide back-channel responses in order to ground their information states or to signal engagement, agreement, and alignment (Yngve, 1970). For example, a listener’s head nod or the use of “yeah” can function as continuer, alignment token, and agreement signal while a “hmm” or a head shake usually signals a weak conversational engagement or disagreement (Lambertz, 2011). In symmetry, speakers have similar ways of signaling their cognitive states in conversation, such as by filling a pause before an answer with “um” or “oh” (Brennan and Williams, 1995; Smith and Clark, 1993). Speakers use fillers such as “um” and “uh” to display the fact that they are working on producing an utterance. Hearers can use this information, that means the presence or absence of different kinds of fillers, to make accurate inferences about the speaker’s commitment to an answer based on the display that precedes the answer (Brennan and Williams, 1995). In this, an unexpected delay licenses the inference that the conversational partner is having difficulty. Finally, an implicit method of providing evidence for the common ground is by simply continuing the conversation with the next relevant utterance, whereas a listener frequently requests clarification of some or all of the contributor’s presentation if an utterance or action has not been understood or accepted.

Referential Grounding

During natural conversations and social joint activities, speakers collaborate with their partners by producing references to entities in the common conversational or perceptual ground. Thus, on the one hand, they use references to the *discourse context*, such as ellipsis and pronominal references to entities, previously mentioned during the conversation, as well as references to the *situated context*, which means to objects, persons, or events in the physical environment (Brennan, 1998). The process of *referential grounding* requires that both partners are responsible for establishing the mutual belief that they both have understood the referring expression and associating the same object or entity with the reference. According to the contribution theory they collaborate to reach this mutual belief, that means, the speaker is looking for reliable evidence of the listener’s understanding while the listener tries to provide this understanding (Clark and Brennan, 1991). There exist various methods that interaction partners can use to jointly coordinate their mutual understanding of references and improve the efficiency of referential grounding with as little effort as possible. Of course, in all types of conversations, including those that do not require a physical proximity of the interaction partners, such as telephone conversations or e-mail exchanges, a clarifying subdialog, in which ambiguities are resolved, can be initiated. In this thesis, I do not specifically focus on the management of clarification dialogs or the grounding of references

REFERENCE
CONTEXT

to the discourse context but rather on the nonverbal behaviors and the gaze mechanisms that contribute to the multi-modal grounding of references to the situated context, that means objects, persons, or events in the shared physical environment.

COLLABORATIVE EFFORT During multi-modal social interactions and joint activities the participants distribute information across different modalities (Oviatt, 2003, 2012). In this, people divide their efforts between vocal utterances, gestural actions, facial expressions, gaze behaviors, material signals, and many more. In principle, they could exclusively rely on the verbal exchange of information, but it is obvious that they do not, as soon as they are not forced to communicate only verbally. In face-to-face interactions, such as situated dialogs and joint activities, people not only speak but also nod, smile, point, gaze at each other and objects in the environment, and exhibit and place things (Clark and Krych, 2004). The main reason that they communicate this way is the principle of minimizing their *collaborative effort* (Clark and Wilkes-Gibbs, 1986; Clark and Schaefer, 1989; Clark and Brennan, 1991; Clark, 1996) which has been defined by Clark and Brennan (1991) as

*“the work that both do from the initiation of each
contribution to its mutual acceptance”*

According to this principle, people are opportunistic in the way they exchange information and achieve a mutual common ground. They always try to select from the available methods the ones they think take the least effort for the two of them jointly. This is for instance measured in terms of cost in time, resources, and error recovery costs (Clark and Brennan, 1991). They exploit a combination of speech, gestures, and other modalities they judge will take the least joint effort with regard to the expressive power of each communication channel as well as the assessment of their partner’s ability and effort to combine information from the different modalities.

SITUATED GROUNDING This multi-modal way of reducing the collaborative effort is especially prominent when producing references to the situated context in face-to-face interactions and collaborative joint activities, like in the illustrative scenario. In general, such a direct face-to-face interaction does not take place in an empty space but is always linked to the current environment and physical context of the interaction. For instance, in the illustrative scenario, the physical context consists of the table and the photos displayed on it as well as Marley’s apartment with the living room, the kitchen and all objects contained therein, such as the kettle in the kitchen or the chairs they are sitting on.

When two interaction partners collaborate on such a physical task on a shared workspace or are discussing objects in the environment, then they frequently transform their language to reduce the collaborative effort and make the communication more efficient. To refer to objects in the environment they distribute information across different modalities, depending on the collaborative effort and the expressive power of each channel, and rely on their partners’ ability to combine this information in order to resolve ambiguities (Gergle *et al.*, 2004;

Oviatt, 2012). For example, Marley and Charly both point and look at photos and are manipulating them in addition to verbally referring to them. They nod and smile at each other to nonverbally acknowledge their reference understanding because they know that their faces and the shared workspace are visible for both.

In face-to-face interaction a very efficient way to coordinate the mutual understanding of references is to share each other's visual information about the physical context of the interaction. Therefore, the partners observe each other's gaze to notice their visual attention (Argyle and Cook, 1976), point to objects or look at them and at the same time use deictic references, such as "that one" or "there", or they demonstrate and manipulate objects in front of their interaction partners (Clark and Krych, 2004; Clark, 2005). An example of such a combination of directed gaze and a verbal deictic reference can be found in scene ③ of the illustrative scenario, when Marley asks for information about a specific photo:

- ③ Suddenly, Marley is asking 'Tell me! Where is this beach?' while looking at one of the shown photos. Then she looks at Charly who returns the gaze and says 'This was your trip through France in 1980!'

In this situation, Marley's verbal referring expression is ambiguous and could basically refer to any photo on the surface table. However, Charly is able to determine the photo that Marley most likely refers to by taking her gaze direction into account. He thinks to have enough evidence that Marley is focusing on exactly that photo so that he can continue with his next contribution by just providing the information that Marley asked for. In this way, he tries to reduce the collaborative effort and at the same time accepts that the common ground will be disrupted and a costly clarification dialog will be required if his guess was wrong.

Research has shown that using visual information to infer what another person knows facilitates efficient communication and reduces the ambiguity that might otherwise be associated with specific linguistic expressions. Monitoring the partner's eye-gaze and head position provides cues that allow tracking each other's perspectives, attentional states, and intentions, without requiring a very memory intensive cognitive model of mutual belief and common ground (Hanna *et al.*, 2003). Following the directed gaze of the partner and sharing the partner's attention to an object in the environment helps to establish a common perceptual ground (Sebanz *et al.*, 2006) and to share a common point of reference (Mundy and Newell, 2007). Interaction partners with a mismatched perception of the shared environment can benefit from shared gaze when repairing the common ground (Liu *et al.*, 2013). Being aware of the partner's eye gaze may enable the nonverbal execution of joint actions (Brennan *et al.*, 2008). Various studies have shown that eye gaze is a powerful and flexible tool that facilitates the disambiguation of speech in referential communication (Hanna and Brennan, 2007). It has been shown that when one partner pays attention to the other partner's gaze during an interaction, grounding of references to objects in the environment becomes more efficient (Liu *et al.*, 2013). This is especially significant when the description of behaviors and actions of the joint task or objects in the shared environment are linguistically complex (Gergle *et al.*, 2004). This is often the case when the tasks are visually complex or when participants have no simple vocabulary for describing their environment (Gergle *et al.*, 2004).

ROLES OF GAZE
FOR GROUNDING

Consequently, it is obvious, that being aware of the partner's eye gaze and visual information plays an essential role in referential grounding in human-computer interaction, for example, when interacting with embodied conversational agents (Nakano *et al.*, 2003) or collaborative robots (Mehlmann *et al.*, 2014a,b) and in computer-supported teamwork (Gergle *et al.*, 2013).

2.1.3 Synergy and Overlap Effects

BEHAVIOR MATCHING VS. INTERACTIONAL SYNCHRONY The distinction between behavior matching and interactional synchrony is sometimes unclear because these two types of coordination are not completely disjoint and are frequently observed simultaneously or in combination. Their precursors are often similar which suggests that both work together *synergistic* to serve the goal of interpersonal coordination in social interactions and joint activities (Schmidt and Richardson, 2008; Marsh *et al.*, 2009). The pro-social effects of being in sync with a partner are similar to those of being mimicked by him. Both, behavior matching and interactional synchrony are interdependent and mutually reinforcing each other, having many similar pro-social effects, such as increasing liking and rapport, perceptions of similarity, and feelings of closeness as well as cooperation and helping behavior (Schmidt *et al.*, 2012; Chartrand and Lakin, 2013).

MUTUAL INFLUENCE VS. ADAPTATION One difference between them is that interactional synchrony, unlike behavior matching, is dynamic in the sense that the important element is the issue of timing, rather than the expression of similarity of behaviors. This suggests that interactional synchrony comprises not only the constant monitoring of the interaction partners' behaviors and a highly reactive response to them, but it might also comprise some degree of anticipation of another person's behaviors, such that the own movements, actions, and behaviors can be closely coordinated in time with those of the partners (Lakin *et al.*, 2003; Marsh *et al.*, 2009). So, on the one side, behavior matching reflects the *mutual influence* of the interaction partners' behaviors, independent of the temporal development of this influence. On the other side, interactional synchrony reflects a more dynamic and reciprocal aspect of coordination which can be considered as the *mutual adaptation* of the interaction partners (Richardson *et al.*, 2005).

For instance, if we consider two people sitting opposite to each other at a shared workspace, as in the illustrative scenario. They are both looking at the same object on the workspace and both are having their head resting on their hand. This static situation certainly exhibits behavior matching in form of the mirroring mechanism or the chameleon effect. However, the mutual coordination and adaptation of their behaviors becomes a matter of timing and synchrony when one of them starts changing the posture or starts looking at another object. In this case, the partner can imitate this behavior and follow the other's attention nearly simultaneously, or within a certain time window in order to achieve a high degree of interactional synchrony.

INTER- VS. CROSS-MODAL MECHANISM Another difference between behavior matching and interactional synchrony lies in the use of different modalities. While both mechanisms can occur in multiple modalities, in behavior matching the adaptation occurs in the same modality for both partners, whereas the reciprocal behavioral patterns that create interactional synchrony can comprise individual behaviors

in completely different modalities (Delaherche *et al.*, 2012).

For example, a listener's back-channel signal which is used during a speaker's verbal utterance to signal agreement or understanding can be produced nonverbally, for example, by nodding with the head or a short smiling expression, which results in a *cross-modal* behavioral pattern. However, it can as well be produced with a short verbal statement, for example, by saying "hmm" or "ahh", which produces an *inter-modal* behavioral pattern. Another example is the direction of the interaction partner's visual attention in the same modality by using directed gaze only, or in another modality by using a pointing gesture, a verbal referring expression, or a combination any of these modalities.

Finally it has been argued that the rich patterns of interpersonal behavioral coordination, especially the coordination of the participants' postural sways and eye movements, reflect the coordination of the underlying cognitive states and processes (Richardson *et al.*, 2008). This coupling between the participants' motor movements is an indicator of both the process and the success of their communication. These findings suggest that there might exist some interrelation between interpersonal coordination and grounding which has however not explicitly been studied. At least the individual social functions of different gaze behaviors and voice activity which are explained in Section 2.2 play a more or less important role and contribute to both interpersonal coordination and grounding in social interactions and joint activities. However, it can be observed, that the behavioral aspects of interpersonal coordination more directly target on specific social outcomes, such as interpersonal rapport, affiliation, connectedness (Bernieri *et al.*, 1994; Lakin and Chartrand, 2003), liking (Hove and Risen, 2009; Miles *et al.*, 2009), compassion, cooperation, altruistic behavior (Wiltermuth and Heath, 2009; Valdesolo and DeSteno, 2011), and emotional empathy (Chartrand and Lakin, 2013), and involve more subconscious, automatic, and reflexive behaviors. In contrast, the mostly conscious and deliberate behaviors that contribute to grounding target on the exchange and coordination of knowledge, intentions, and beliefs for the maintenance of the common ground and, thus, focus more on the efficiency and success of the conversation or task (Hanna *et al.*, 2003; Sebanz *et al.*, 2006; Hanna and Brennan, 2007; Mundy and Newell, 2007; Brennan *et al.*, 2008; Liu *et al.*, 2013).

INTERPERSONAL
COORDINATION
VS. GROUNDING

For example, on the one hand, the aspects of rhythm and simultaneity when following the interaction partner's eye movements certainly contribute to interactional synchrony and thus to rapport and connectedness. On the other hand, the effect of sharing the visual attention and thus the same perceptual ground contributes to grounding in the sense that it facilitates the prediction of the partner's intentions and multi-modal language understanding. Another example is the proper use of gaze cues and monitoring the partner's gaze to enable a seamless exchange and smooth meshing of speaker turns. This certainly contributes to interactional synchrony and can improve cooperation and liking, but, it certainly also helps to ensure an effective and successful coordination of the conversation's process and progress by grounding the speaker and listener roles.

2.2 The Different Functions of Gaze Behavior

Humans communicate via multiple parallel information channels and modalities to achieve interpersonal coordination and grounding with the least collaborative effort. They use speech acts but their voice also reveals information via paralinguistic features, such as prosody, pitch, volume, and intonation. Furthermore, they unconsciously exhibit information via physiological cues such as blood circulation and breathing patterns. Finally, they make extensive use of gestures and postures as well as facial expressions and eye gaze behaviors. Among those, especially *gaze behavior* and its interplay with the other modalities plays versatile roles and contributes to interpersonal coordination and grounding with various functions that can be categorized according to a handful of social contexts (Srinivasan and Murphy, 2011; Jokinen *et al.*, 2013; Fischer *et al.*, 2015; Ruhland *et al.*, 2015; Mehlmann *et al.*, 2014a,b, 2016). Besides the general objective to establish agency and liveliness, it plays a role in the expression of social attention and intention, the support of interaction content, language understanding and recall, the regulation of the interaction process and participant roles, the communication of emotional and cognitive states, and the creation of social closeness and rapport as well as the regulation of intimacy. Following this categorization, I now present relevant research and literature on the different roles of gaze behaviors for social and regulatory functions that contribute to interpersonal coordination and grounding.

2.2.1 Concepts and Definitions

Humans exploit a variety of interactional *gaze mechanisms* with different social functions and manipulate a variety of temporal and spatial variables, and behavioral parameters of gaze to achieve certain social outcomes. These variable features of their gaze behavior, such as frequency, duration, timing, and the direction or target they look at vary with the context and type of the activity as well as the distribution of *participant roles*. In the following, I present the most important interactional gaze mechanisms and features as well as participant roles that can be found in social joint activities. In order to avoid unclarity or misunderstandings due to conceptual and terminological confusions, I therefore mainly rely on the categorizations by von Cranach and Ellgring (1973a,b) and Goffman (1979).

Gaze Mechanisms

VISUAL ORIENTATION Figure 2.2.1 shows an illustration of important gaze mechanisms in social interactions. The terms *directed gaze* and *visual orientation* refer to situations in which the sender of gaze looks at the eye region or the upper half of the recipient's face (Cook and Smith, 1975; Argyle and Cook, 1976) but also when looking at an object, event, or person in the environment or point in space (Figure 2.2.1 Ⓐ). A person's gaze direction or target can be an indicator of particular interest in a specific object or interaction partner (Gibson and Pick, 1963). Directed gaze movements that are supposed to indicate attention, suggest future actions, and define the target of facial signals, for example, when multi-modally referring to an object or person in the environment, are often called *deictic gaze* (Argyle and Cook, 1976; Shepherd, 2010).

The concepts *mutual gaze* or *eye contact* describe situations in which both partners look into each other's face or eye region, at the same time, and both are aware of this mutual visual orientation (Figure 2.2.1 (B)) (von Cranach and Ellgring, 1973a,b). Being looked at, or having even the feeling of being monitored by another person, can cause certain physiological and neurological reactions (Coss, 1970; Pelfrey *et al.*, 2004), such as an involuntary pupil dilation as indicator for increased emotional arousal. Eye contact can have various effects on the social framework and interpersonal relationship, such as increased attention (Langton *et al.*, 2000) and intimacy (Argyle and Dean, 1965).

MUTUAL
ORIENTATION

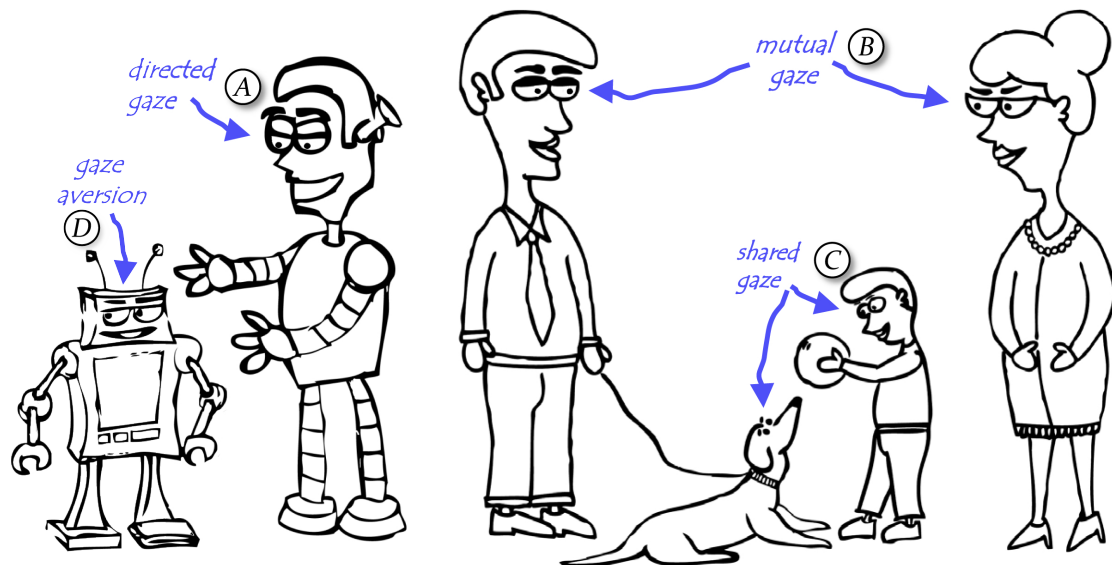


Figure 2.2.1: An illustration of some of the most important gaze mechanisms in social interactions.

The term *gaze following* means that one partner follows the other's line of sight to a point in space or object in the environment (Emery, 2000). Vice-versa *gaze direction* means the successful attempt of one partner to direct the other's gaze to the point in space. If both partners then look at the same fixed point in space they have established *shared gaze* (Figure 2.2.1 (C)), also referred to as *shared attention* (Butterworth, 1991; Emery, 2000). If both interaction partners are not only simultaneously looking at each other or the same object, person, or event in the environment, but, also cognitively attending them intentionally at the same time and being mutually aware of this connection, then this is referred to as *joint attention* (Clark and Marshall, 1981; Baron-Cohen, 1995; Tomasello, 1995; Tomasello *et al.*, 2005). In human-agent interaction research, shared and joint attention are often confused or used interchangeably. State-of-the-art behavior and interaction models cover solely individual aspects of shared attention and cannot capture true cognitive joint attention. As in this thesis too, they actually research the situation in which the partners are visually perceiving the same object, person, or event without requiring commutated mental states (Kaplan and Hafner, 2006; Pfeiffer-Lessmann *et al.*, 2012).

VISUAL
ATTENTION

The terms *averted gaze*, *gaze avoidance* or *gaze aversion* describe a situation in which a person avoids looking at or is looking away from the partner (Figure 2.2.1 ④), especially if already being looked at (von Cranach and Ellgring, 1973a; Emery, 2000). The formulation *mutual gaze aversion* means that both partners simultaneously avoid being looked at by the other. In contrast to visual orientation, gaze during gaze aversion targets fixation points and random locations in the environment or points on a partner's body but not the face. These targets can change based on conversation structure and content, the objects of interest and the relevance of these objects to the joint activity (Argyle and Graham, 1976).

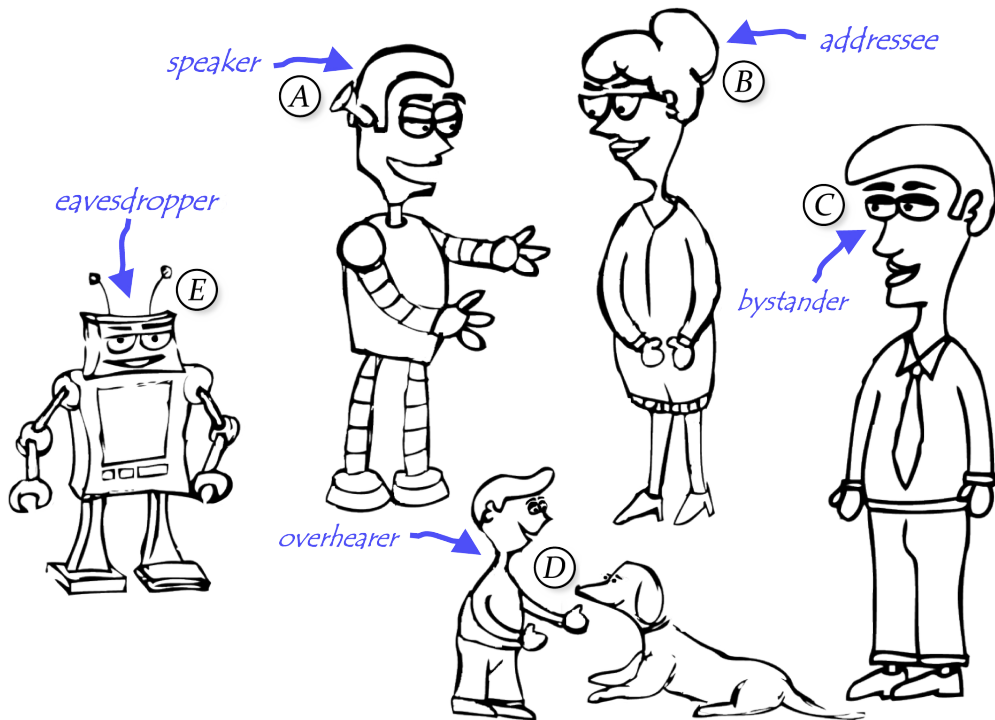


Figure 2.2.2: An illustration of some of the most important participant roles in social interactions.

Participant Roles

Figure 2.2.2 shows an illustration of the different roles of participation of a social interaction, as described by Mutlu *et al.* (2009) who adapted them from Goffman (1979) and Clark (1996). Traditionally, each participant either plays the role of the *speaker* (Figure 2.2.2 ①) or the *addressee* (Figure 2.2.2 ②) at any moment during a *dyadic conversation* (Goffman, 1979). This usually depends on the partners' speech activities and if they are presenting or accepting a contribution (Clark, 1996). While the primitive notions of speaker and hearer are commonly considered inadequate for other *two-party interactions* there is, however, no agreement on a common scheme of the different producer and receiver roles (Goffman, 1979; Levinson, 1988; Clark and Schaefer, 1992). In this thesis, it is assumed that in a *collaborative joint activity*, an *actor*, which is a participant that performs an action on the shared workspace, such as manipulating an object, can be put on a level with the speaker role, if a turn or contribution can also be achieved with such an action (Clark, 2005). How these roles shift during interactions is considered important in understanding spoken discourse and joint activities, for

example, when grammatical choice is determined by the participant constellation (Hymes, 1972; Hanks, 1996). Also interesting is how people use their voice activities, gaze behaviors, and other nonverbal signals to shape and reciprocally exchange their participant roles.

Multi-party interactions can additionally include different *side-participant* roles as well as more or less ratified or acknowledged *non-participants* (Goffman, 1979; Wilkes-Gibbs and Clark, 1992; Clark, 1996). Special side-participants are *listeners* which are actively listening but are not directly addressed by the speaker and can become addressees or speakers at any time during the interaction. People having the role of *bystanders* (Figure 2.2.2 ©) are attentively observing and listening to the conversation without actively participating while, nevertheless, being acknowledged by the participants of the conversation (Goffman, 1979; Clark and Carlson, 1982; Clark, 1996). Those that have not been acknowledged as participant are *overhearers* (Figure 2.2.2 Ⓓ) when unintentionally, or *eavesdroppers* (Figure 2.2.2 Ⓔ) when purposefully listening to the conversation (Goffman, 1979).

2.2.2 Attention and Intention

An important aspect of interpersonal coordination and prerequisite for grounding is the ability to orient one's own and to direct another person's attention to information in the environment that is relevant to one's own behavioral goals and intentions (Posner, 1980; Frischen *et al.*, 2007). One of the most frequently used and reliable methods to share attention and predict the interest of the interaction partner in a specific object, person, or event is gaze following. Humans usually follow their partners' gaze shifts and movements to share their partners' visual focus and point of reference (Mundy and Newell, 2007). This usually results in recurring phases of shared visual attention to these objects or points in space (Kendon, 1967). By following their interaction partners' gaze shifts people also signal that they are engaged and interested in the joint social activity and are able to identify referred objects of interest which finally helps in the maintenance of the common ground (Clark, 1996, 2005).

The whole time during a joint activity, the interaction partners attentively observe each other's actions and behaviors, trying to predict and anticipate each other's intentions and objectives, in order to adjust their own behaviors accordingly (Sebanz and Knoblich, 2009; Huang *et al.*, 2015). Among other cues, gaze direction has been identified as crucial in understanding the intention of the interaction partner because it may not only be an indication for cognitive attention and interest but also of the actions and steps that they could subsequently perform (Baron-Cohen *et al.*, 2001; Meltzoff and Brooks, 2001). For example, the partners would reasonably assume and agree that an area or object in the physical environment, being jointly gazed towards, will probably be the next space or entity to be acted upon during the joint activity (Baron-Cohen *et al.*, 2001; Meltzoff and Brooks, 2001). Furthermore, monitoring the partner's gaze direction can help to faster detect situations in which the partner needs support, thus being able to provide faster help when it seems necessary (Brennan *et al.*, 2008). Thus, being aware of and sharing the partner's gaze facilitates an effective task coordination (Tomasello, 1995) and maintaining the common ground to ensure the success of the joint collaborative task (Clark, 2005).

ATTENTION
DIRECTION

Besides verbal references and deictic gestures, humans use directed gaze, sometimes in combination with the other modalities, to intentionally or unconsciously direct their partners' attention to objects, events, or points in space (Baron-Cohen, 1995; Emery, 2000), or to themselves (Clark and Wilkes-Gibbs, 1986; Bangerter, 2004; Richardson and Dale, 2005; Oviatt, 2012). They direct their gaze at a partner to signal attention to the other person or, vice-versa, avert the gaze in order to avoid to draw attention (Goffman, 1963). Being looked at is a social stimulus that triggers a neuro-physiological attention detection mechanism (Baron-Cohen, 1995; Perrett and Emery, 1994) which is frequently responded to with increased arousal (Nichols and Champness, 1971; Patterson, 1976; Kleinke, 1986). When simultaneously paying attention to each other, then people usually perform mutual gaze (Argyle *et al.*, 1973; Argyle and Cook, 1976) which has been found to be a clear indication of engagement in the interaction and attentiveness towards the partner (Sidner *et al.*, 2005). When a joint activity requires complex references to, and joint manipulation of objects in the physical environment, then they, however, more frequently look at these objects of shared interest than to each other in order to signal engagement (Argyle and Graham, 1976; Anderson, 1997; Nakano and Ishii, 2010). Drawing the interaction partners' attention and interest towards an intended object helps to establish a common perceptual ground (Sebanz *et al.*, 2006) and thus facilitates the grounding process (Clark, 1996, 2005).

2.2.3 Understanding and Recall

MULTI-MODAL
DISAMBIGUATION

Once a common perceptual ground has been established, gaze also serves the understanding of multi-modal references to focused objects. Humans usually distribute information across modalities, often a combination of directed gaze and a verbal referring expression or deictic gesture, to produce multi-modal references (Oviatt and Cohen, 2000; Oviatt, 2003, 2012). In return, they rely on their partners' ability to take their gaze behavior into account for the reliable resolution of this reference (Oviatt and VanGent, 1996; Oviatt, 1999, 2002, 2003). Indeed, the speaker's gaze direction rapidly constrains the domain of interpretation for an addressee and thus speeds up the resolution process (Hanna and Brennan, 2007). This can efficiently reduce the collaborative effort (Clark and Wilkes-Gibbs, 1986) compared to using lengthy verbal descriptions of an object's attributes and relations to other objects (Oviatt, 1996, 1997). For that reason, the disambiguation of ambiguous references with gaze is frequently used to ground the interaction content instead of engaging in lengthy clarification dialogs (Oviatt *et al.*, 1997, 2000).

CO-VERBAL
ALIGNMENT

It has been found that directed gaze to an object during a multi-modal referring statement is temporally well-aligned with corresponding referring expressions in verbal utterances (Meyer *et al.*, 1998; Griffin and Bock, 2000; Griffin, 2001). Some research shows, that referential gaze typically precedes the corresponding referring expression in speech by roughly 800 to 1000 milliseconds (Meyer *et al.*, 1998; Griffin and Bock, 2000; Griffin, 2001) and listeners typically take 500 to 1000 milliseconds to fixate an object after the linguistic reference (Allopenna *et al.*, 1998). Similar work shows that a gaze fixation that best identifies a referred object, on average, occurs about 600 milliseconds before speech onset with a range of 150 to 1200

milliseconds for individual subjects (Kaur *et al.*, 2003). These results show, that, depending on the application, task, or subject, certain temporal alignment and integration rules can be applied for multi-modal disambiguation and early repair of misunderstandings (Richardson *et al.*, 2007a). For example, the listener's gaze following about half a second after the speaker's verbal reference can be pulled up to determine if the listener successfully identified the meant object. In case of a misunderstanding, the speaker can immediately repair the common ground by refining or rephrasing the description or by engaging in a clarification dialog in which the remaining ambiguities are resolved.

Finally, it has been shown that the frequent maintenance of mutual visual orientation between two participants has a positive effect on language understanding and learning due to a better recall of the conversation's content (Exline and Eldridge, 1967). It has been shown that consciously well-timed eye contact between a teacher and the students can improve the students' recall of the lecture content (Otteson and Otteson, 1980; Sherwood, 1987) and their task performance (Fry and Smith, 1975), and increases the efficacy of the lecture as a whole (Brooks, 1985; Woolfolk and Brooks, 1985). This effect has been explained with the higher arousal and concentration during the feeling of being looked at (Nichols and Champness, 1971; Patterson, 1976; Kleinke, 1986).

RECALL &
LEARNING

2.2.4 Turn-Taking and Feedback

Interpersonal coordination and grounding includes a smooth interaction flow through the efficient and coordinated regulation of the participant roles (Kendon, 1967; Duncan, 1972; Clark, 1996). In this, gaze plays important roles in *footing* (Goffman, 1979; Clark, 1996) and *shifting* (Nielsen, 1962; Kendon, 1967; Duncan, 1972, 1974; Sacks *et al.*, 1974; Goodwin, 1980, 1981) these roles when handling interruptions and negotiating turn-exchanges at overlapping talk and pauses (Schegloff, 2000, 2001; Goodwin, 1980, 1981). It plays a monitoring as well as a regulating role for the floor management at transition-relevant places (Sacks *et al.*, 1974), such as mid utterance hesitations (Bavelas *et al.*, 2002) and the end of phrases.

TURN-TAKING
REGULATION

A variety of multi-directional and multi-modal turn-regulation patterns require the precise alignment of gaze with other modalities, such as speech and gestures (Kendon, 1967; Duncan, 1972; Clark, 1996). For example, speakers usually look away from their addressees shortly after taking the floor and when they begin talking in order to indicate that they want to keep the floor (Nielsen, 1962; Duncan and Fiske, 1977; Cummins, 2012). They avoid mutual gaze when they do not want to be interrupted during speech, want to retain their speaker role during speech hesitations, and while constructing their next utterance (Kendon, 1967; Ho *et al.*, 2015). Furthermore, speakers look at an addressee to signal the end of a contribution and to propose that participant as the next speaker (Nielsen, 1962; Kendon, 1967). The turn exchange can be delayed if a contribution does not end with such a mutual gaze (Kendon, 1967; Vertegeal *et al.*, 2000). Not only speakers are responsible for turn-taking but also listeners, for example, when making more gaze shifts prior to speaking as a way to request the turn (Harrigan, 1985). It has been shown that gaze becomes all the more important in signaling role exchanges in conversations between strangers (Beattie, 1980).

TURN-TAKING
SIGNALS

MULTI-PARTY
TURN-TAKING

When addressing a group of people, then speakers usually tend to rather evenly distribute their gaze among its members. Holding eye contact with an individual member for long enough can then be a signal that the speaker is addressing or yielding the turn to this particular participant (Bales, 1970; Sacks *et al.*, 1974; Goodwin, 1981) who is then considered as main target of the current communication content (Bales *et al.*, 1951). Breakdowns in the organization of multi-party conversations can thus occur if the speaker does not clearly signal who is being addressed at the end of his turn (Schegloff, 1968). Usually, the intended addressee answers the speakers gaze, but, if he apparently and intentionally avoids mutual gaze, then the speaker can feel excluded or ignored (Williams, 2001). Humans are very sensitive to the slightest cues of gaze for inclusion or exclusion from a group. A simple eye contact is sufficient as acknowledgment conveying group inclusion while withholding eye contact can signal exclusion (Wesselmann *et al.*, 2012).

BACK-CHANNEL
ELICITING

While gaze behaviors usually correlate with turn transitions and role shifts they are also observed in conjunction with *back-channel eliciting cues* (Sandgren *et al.*, 2012). Generally, the participants of an interaction regularly produce back-channel signals using nonverbal cues such as head nods or short verbal statements. Listeners usually use back-channel cues to signal, for example, understanding, agreement, or engagement (Yngve, 1970; Allwood *et al.*, 1993) without requesting the turn while the partner is speaking or performing an action. They shortly acknowledge that they have understood what has been said and done to ground their information states without the need for a costly interruption of the conversational flow and a new negotiation of the participant roles (Yngve, 1970).

In return, speakers occasionally perform a short glance of mutual gaze to the addressee without yielding the turn with the aim to elicit a back-channel cue at specific points in time (Kendon, 1967; Allwood *et al.*, 1993; Bavelas *et al.*, 2002). For example, Oertel *et al.* (2012) and Hjalmarsson and Oertel (2012) showed that a frequently occurring gaze cue shortly before a back-channel is a gaze glance to the listener initiated up to 2 seconds before the onset of the actual occurrence of the back-channel. In line with the findings of Kendon (1967) and Bavelas *et al.* (2002), they showed that gaze is a back-channel inviting cue, in the sense that back-channels are indeed associated with an increase in mutual gaze directly preceding the onset of the feedback expression. They explain this observation with the fact that listener responses often are visual responses, such as eyebrow raises, head nods, or a smile, with or without an accompanying verbal back-channel. In order to detect these visual nonverbal cues, the speaker has to look at the listener and since these kind of responses are usually very short they lead to a very short glance of mutual gaze. Although the gaze cues for feedback eliciting and turn-yielding might look very similar at first sight, it is very important not to confuse these signals and to handle them differently (Duncan and Niederehe, 1974).

2.2.5 Emotions and Cognition

EMOTIONAL
DISPLAYS

Interpersonal coordination and grounding involves being sensitive toward the interaction partners' cognitive and emotional displays while adapting one's own behavior in response and producing adequate verbal and nonverbal cues to reveal the own thoughts and feelings

(Kendon, 1967; Argyle and Cook, 1976; Doherty-Sneddon and Phelps, 2005). Humans usually convey emotional states via a variety of modalities, such as facial displays and postures that are frequently enhanced by gaze behaviors. For example, avoidance-oriented emotions, such as fear, are typically accompanied by averted gaze while directed and mutual gaze, is linked to approach-oriented emotions, such as joy (Adams and Kleck, 2005). Moreover, research suggests that gaze aversion is an indicator for nervousness that could be associated with deception (Zuckerman and Driver, 1985; Vrij, 2002). Others found that it is associated with an increase in anxiety and plays a role in anxiety reduction (Stanley and Martin, 1968) or that people engage in less eye contact when they feel embarrassment (Exline *et al.*, 1965).

Understanding and sharing each other's emotional states increase engagement and rapport in an interaction (Chartrand and Bargh, 1999). A simple form of emotional grounding can be realized by mimicking the emotional cues of the conversational partner whenever the partner is trying to establish eye contact (Chartrand and Lakin, 2013). Such ideomotor behaviors may then convey the impression of empathy even in the absence of any understanding or reappraisal of the other's emotional state and the underlying reasons (Hess and Fischer, 2013, 2014). Real empathy, however, requires a deeper level of mind reading (Baron-Cohen, 1997) from the partner. This means the partner has to appraise the situation from the perspective of the conversational partner to be able to produce a sensitive and adequate response.

EMOTIONAL
GROUNDING

Gaze cues are also among the key signals to display cognitive states and operations. People look away from their interaction partners because gaze aversions reduces visual stimulation from their faces (Ro *et al.*, 2001) and thus supports cognitive activity (Ehrlichman and Miccic, 2012). It facilitates disengaging from the environment and partners and, thus, to limit visual inputs and avoid mutual gaze that could interfere with the production of speech (Beattie, 1980, 1981b). For example, speakers usually avert gaze when planning speech (Kendon, 1967), signaling cognitive effort (Argyle and Cook, 1976), updating beliefs, desires, and intentions (Doherty-Sneddon and Phelps, 2005), or when they encounter a rejection or counterproposal from their conversation partner and want to avoid a conflict or threat (Argyle and Cook, 1976). People look at objects while updating their belief about or planning an action on these objects (Argyle and Cook, 1976). Forcing oneself to look away from the conversational partners while recalling information from long-term memory or when planning a response to a challenging question significantly improves performance (Glenberg *et al.*, 1998; Doherty-Sneddon and Phelps, 2005; Phelps *et al.*, 2006). The listeners, in turn, are frequently showing curiosity, engagement, understanding, and attentiveness by responding to these gaze behaviors by trying to establish mutual gaze (Sidner *et al.*, 2005; Bee *et al.*, 2010a) and producing appropriate acoustic and visual attentive behaviors and back-channel signals (Kendon, 1967; Yngve, 1970; Allwood *et al.*, 1993; Bavelas *et al.*, 2002). Another cognitive operation that produces particular gaze behaviors is the monitoring for events and the reaction to unexpected events in the environment. For instance, people frequently respond to unexpected events, such as a loud or unusual sound in the environment, by looking into the direction of the event before continuing the interaction (Yantis and Jonides, 1990; Yantis, 1993; Chopra-Khullar and Badler, 1999; Lee *et al.*, 2007).

COGNITIVE
ACTIVITY

2.2.6 Personality and Intimacy

PERSONALITY EVALUATION An interaction partner's gaze behavior is taken into account when evaluating his or her personality (Goffman, 1963; Kleck and Nuessle, 1968; Kendon and Cook, 1969; Cook and Smith, 1975). Thus, gaze has a significant influence on the reciprocal perception and social attitude towards each other (Argyle *et al.*, 1971). The establishment and maintenance of mutual visual orientation often leads to positive social effects such as social attention (Langton *et al.*, 2000) and increased attraction (Exline and Winters, 1966). For example, up to a certain degree, an interaction partner's personality is generally rated the better the more he looks at his or her partners and responds to the partner's gaze (Argyle *et al.*, 1971). It has been found that people who look at others only about fifteen percent of the time are perceived as cold, pessimistic, cautious, nervous, defensive, immature, evasive, submissive, indifferent, sensitive, and lacking confidence. In contrast, those who look at others about eighty percent of the time are rated as more friendly, self-confident, sincere, and generally more natural (Kleck and Nuessle, 1968; Cook and Smith, 1975).

Besides the pure total amount of gaze, the rhythm of gaze patterns and gaze movements is another evaluation criterion for the perception of the interaction partners. People who have a slower rhythm of gaze shifts, looking in long, infrequent gaze intervals are usually preferred over those who's gaze behavior shows a faster rhythm, looking in short and frequent gaze intervals (Kendon and Cook, 1969). People who are showing attentional engagement by constantly trying to establish eye contact and mutual gaze are perceived as more likable than those who frequently break eye contact and thus showing less attention or engagement (Mason *et al.*, 2005).

INTIMACY REGULATION The proper gaze behavior can also support a positive experience of interpersonal intimacy (Patterson, 1976; Duncan and Fiske, 1977), if carefully used together with other mechanisms to keep the intimacy in balance over the course of an interaction (Abele, 1986). This means, that an increased amount of gaze towards the partner produces positive social outcomes, such as closeness, attention, attraction, and intimacy only to a certain degree. Too much gaze or even staring can lead to the contrary and result in discomfort in the partners due to an excessive and unpleasant degree of intimacy. In this, the participants of a conversation look more at their partners when they are speaking about intimate topics (Exline *et al.*, 1965) and adapt to changes in the topic's intimacy level by increasing mutual gaze while talking about intimate topics and decrease it when talking about non-intimate topics (Abele, 1986).

So, gaze behaviors, such as mutual gaze and gaze aversion, are intentionally and subconsciously used for the regulation of interpersonal intimacy over the course of a social interaction. In this, they interplay with a number of other interactional mechanisms for intimacy regulation such as physical proximity, the intimacy of the conversational content, or the amount of smiling (Argyle and Dean, 1965). For example, smiling and the amount of gaze are inversely correlated (Kendon, 1967) and people reduce gaze toward their partners at closer distances (Argyle and Dean, 1965). By these means, the interaction partners usually aim to develop an equilibrium of intimacy and try to compensate for an uncomfortable and

inadequate increase of intimacy (Argyle and Dean, 1965). Thus, when one of the influencing factors changes, for example, if there can be perceived an increase in physical proximity, people tend to maintain the equilibrium by shifting one or more of the other components in the reverse direction, for example, by averting gaze. In this, intimacy-regulating and floor-managing gaze aversion is more likely to be directed sideways, while cognitive gaze aversions is frequently directed upwards (Andrist *et al.*, 2014).

2.3 The Functions of Overlaps and Interruptions

Speech *overlaps* and *interruptions*, whether produced intentionally or subconsciously, are omnipresent and frequently observed characteristics of conversational speech. Besides demonstrating activeness and liveliness, their functions are the expression of social attitudes and intentions as well as the regulation of the interaction process. Especially, interruptions are important for interpersonal coordination and grounding due to their impact on the organization of turn-taking and topic changes (Meltzer *et al.*, 1971; Sacks *et al.*, 1974; Zimmerman and West, 1975; Bennett, 1981; Drummond, 1989; Olbertz-Siitonen, 2009; Tannen, 2012; Levinson and Torreira, 2015). In the following, I present relevant research and literature on the different roles of speech overlaps and interruptions for social and regulatory functions that contribute to interpersonal coordination and grounding.

2.3.1 Concepts and Definitions

Even though, speech overlaps and interruptions have drawn much attention from linguists and have been extensively investigated, there is still no agreement on their generally accepted definitions. Discrepancies can also be found between the mostly incomplete and context-specific taxonomies for the two conversational phenomena. These coding schemes distinguish to varying extent and point out different relationships between overlaps and interruptions. They can rather serve as theoretical frameworks for the annotation and retrospective analysis of conversations but are to a lesser extent applicable to computational behavior and interaction models for social agents in real-time interactive systems.

Overlaps and Interruptions

Many definitions do not distinguish between interruptions and overlaps. For example, Meltzer *et al.* (1971) straightforwardly define an interruption as a situation in which two persons are vocalizing something at the same time. Zimmerman and West (1975) more operationally define interruptions as a turn of the next speaker that starts during, but, at least two syllables after, the beginning or before the end of the current speaker's turn. A similarly mechanical definition is provided by Esposito (1979), who defines that an interruption occurs whenever one speaker cuts off more than one word of another speaker's utterance. Leffler *et al.* (1982) operationally define interruptions as situations in which one subject says at least two consecutive identifiable words or three syllables of a single word while another is speaking.

*EQUATING
DEFINITIONS*

A more differentiated definition is that of [Feldstein and Welkowitz \(1987\)](#), who divide simultaneous speech descriptively into *interruptive* and *non-interruptive*. Non-interruptive is an overlap if it starts and ends while the participant with the speaker role unswervingly continues talking. Interruptive simultaneous speech begins while the person who has the floor is talking and ends after he has stopped. There also exist more interpretative and context-dependent definitions of the two phenomena. For example, in view of gender differences in discourse, [West and Zimmerman \(1983\)](#) describe interruption as an instrument to demonstrate power, exercise control, and violate the speakers' turn. Finally, [Schegloff \(1987\)](#) provides the probably most appropriate definition for our purpose by defining an interruption as “a violation of the turn exchange system” and an overlapping as “a misfire in it”.

Categories and Taxonomies

Most of the above definitions think of interruptions as involving simultaneous speech. While some analysts consider any speech overlap as an interruption ([Wiens et al., 1965](#)), others regard interruption as a subcategory of overlap ([Kennedy and Camden, 1983b](#); [Dindia, 1987](#)). However, both views are problematic, because not every overlap may be interpreted as interruption ([Drummond, 1989](#)) and interruptions do not necessarily involve overlaps ([Murray, 1985](#)). For example, if two speakers simultaneously start speaking, then an overlap occurs but it is not possible to make any of them responsible for an interruption ([Drummond, 1989](#)). [Murray \(1985\)](#) claims that “simultaneous speech is neither necessary nor sufficient for the recognition of interruption by interlocutors”. For example, the interrupter can start speaking during a short speech pause between two words or sentences in the middle of the speaker's turn. In this case, the speaker could give up the turn immediately before continuing to speak ([James and Clarke, 1993](#)). As shown in [Figure 2.3.1](#), an interruption that takes place without any voice overlaps is also denoted as *silent interruption* ([Ferguson, 1977](#); [Beattie, 1981a](#)).

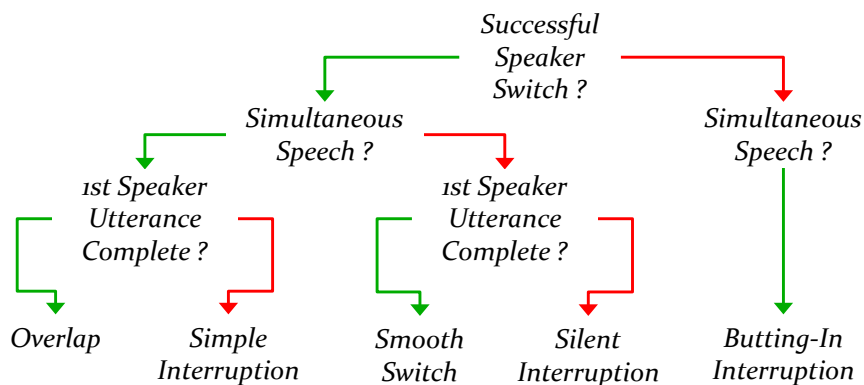


Figure 2.3.1: The types of speaker switch attempts that [Beattie \(1981a\)](#) adopted from [Ferguson \(1977\)](#).

Early categorizations define a successful interruption as an interruption attempt that causes a speaker switch whereas he retains possession of the floor after an unsuccessful interruption ([Jaffe and Feldstein, 1979](#); [Meltzer et al., 1971](#); [Natale et al., 1979](#)). Similar, [Clancy \(1972\)](#) distinguishes between two types of speaker switches comprising speech overlaps, first, those causing broken-off, unfinished sentences of the interrupted speaker and, second, those in

which the previous speaker completes his sentence while the next speaker has already begun his utterance. Going beyond these simplistic views, Beattie (1981a) elaborates the taxonomy shown in Figure 2.3.1 (Ferguson, 1977), which includes three types of interruptions and a single overlap type. It distinguishes between successful and unsuccessful attempted speaker switches. These are then further divided based on the presence of simultaneous speech and the first speaker's utterance completeness. In this, successful means that the interrupter gains the floor whereas he stops speaking before during an unsuccessful attempt.

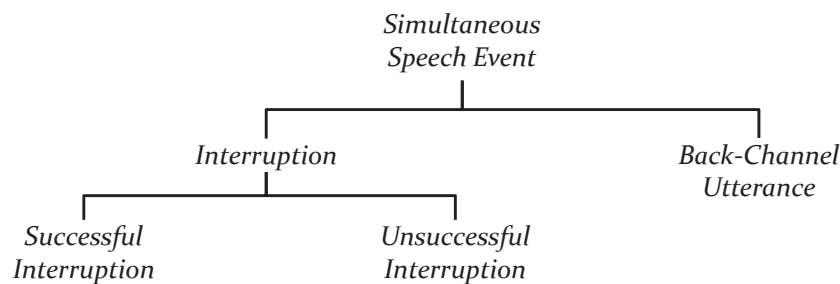


Figure 2.3.2: The types of simultaneous speech events as presented in Roger and Schumacher (1983).

The taxonomy in Figure 2.3.1 does not consider non-interruptive overlaps caused by back-channels or minimal responses (Tannen, 1984, 2012). Therefore, Roger and Schumacher (1983) proposed the classification shown in Figure 2.3.2 which divides simultaneous speech into back-channels and interruptions which are further separated into successful and unsuccessful ones. In *successful interruptions*, the second speaker takes the floor and prevents the first speaker from completing his utterance. In *unsuccessful interruptions*, the second speaker fails to obtain the right to speak, comparable to *butting-in interruptions* in Figure 2.3.1. The scheme in Figure 2.3.2 was later refined into even more distinguishable events, such as false starts, afterthoughts, listener responses, overlaps, and more (Roger et al., 1988).

SUCCESSFUL &
UNSUCCESSFUL

Appraisal and Occurrence

If an overlap or speech activity is interpreted as an interruption attempt and how an interruption affects a person depends on a variety of individual and interpersonal factors (Sacks et al., 1974; Tannen, 1994), such as the personal, social, and cultural background, and conversation style used among the interaction partners (Tannen, 1984, 1994, 2012). In addition, a speaker's right to complete a turn depends on the context, for example, the authority to speak on particular topics or the length and frequency of his or her preceding speech (Tannen, 2012). Therefore one can basically not identify any universally valid syntactical or acoustical interpretation criteria that show the occurrence of an interruption attempt (Murray, 1985).

INFLUENCE
FACTORS

In the sense of this interpretative and context-sensitive perspective, Li (2001), for example, differentiates successful interruptions into intrusive, cooperative, and other categories. Other researchers distinguish between very similar types of aggressive and cooperative interruptions (Kennedy and Camden, 1983a; Tannen, 1994; Murata, 1994), for example, supportive and disruptive interruptions (Ng et al., 1995), less conflicting versus conflicting interruptions

APPRAISAL
DIMENSIONS

(Bennett, 1981), rapport-oriented against power-oriented interruption types (Goldberg, 1990), or with positive, neutral, and negative affective character (Smith-Lovin and Brody, 1989).

OCCURENCE SITUATIONS It has however been found that overlaps and interrupts are not happening at random moments during a conversation. In contrast, they can be observed in conjunction with particular events in the foreground speech. For example, Shriberg *et al.* (2001) observed in different corpora of natural situated and non-situated multi-party conversations, that most interruptions occur at the boundary of speech pauses while a smaller part occurs during continuous speech but tend to be associated with the end of certain word-level events in the speaker's turn, such as back-channels (e.g. "uh-huh" or "mm"), coordinating conjunction (e.g. "and" or "but"), discourse markers (e.g. "well" or "now"), filled pauses (e.g. "uh" or "um") and disfluencies, like repetitions, repairs, and false starts.

2.3.2 Turn-Taking and Feedback

THE IDEALISTIC SITUATION The organizing of the participant role exchange during a social interaction, also referred to as *turn-taking* (Duncan, 1972; Sacks *et al.*, 1974), plays a significant role for interpersonal coordination and grounding because it comprises the coordination of verbal and nonverbal behaviors when timing the alternating flow of speech in a conversation or a collaborative joint activity (Bernieri and Rosenthal, 1991; Clark, 1996). According to the basic *SSJ* model of turn-taking (Sacks *et al.*, 1974), also referred to as "no gap - no overlap" model, dyadic conversations follow an idealistic protocol prescribing that only one person may be speaking at any time and the turn is exclusively exchanged without any delay at *transition-relevant places* (Sacks *et al.*, 1974). The speaker sends nonverbal and verbal signals to the addressee when a turn change is permitted or desirable while the listener waits for these signals to take the turn (Duncan, 1972; Sacks *et al.*, 1974; Orestrom, 1983; Roger *et al.*, 1988).

THE REALISTIC SITUATION In realistic conversations and joint activities, however, people regularly violate this rule by speaking simultaneously, whether with cooperative or competitive intention. They perform minimal responses and verbal back-channels to signal activeness, attention, interest, enthusiasm, or support, or aggressively barge into the partners' turn with the intention to grab the floor and dominate the conversation. The resultant voice overlaps represent an infringement of the *SSJ* model's idealistic turn-taking rules. Thus, especially interruption attempts, are a serious challenge for the organization of the interaction and participant roles. They must consequently be resolved (Roger *et al.*, 1988; Sacks *et al.*, 1974) using resolution devices for negotiating these roles during overlapping talk and pauses (Schegloff, 2000, 2001). So, the insufficient *SSJ* model is only a baseline on which more sophisticated turn-taking strategies including, shared turns and overlapping speech, must be developed (Edelsky, 1981; Coates, 1994; Tannen, 1994). These must also take the role of nonverbal signals into account (Power and Martello, 1986), in particular gaze behaviors (Kendon, 1967) and gesticulation (Duncan, 1972), but also specific manipulation actions on objects in collaborative activities (Clark, 2005) that can have a similar effect on turn-taking as speech (Mehlmann *et al.*, 2014b).

A speech overlap can be the result of cooperative feedback (Yngve, 1970; Allwood *et al.*, 1993) or can unintentionally be produced when the listener tries to get the turn trouble-free during a pause or another possible completion point of the speaker's turn (Sacks *et al.*, 1974). However, an interruption is usually intentionally used to change the topic or acquire the right to speak while the interrupter is totally aware that the speaker might not intent to relinquish the turn (Bennett, 1981; Schegloff, 2000). Thus, an overlap is considered as a neutral state while an interruption is associated with a negative connotation. Bennett (1981) emphasizes that the term "overlap" is *descriptive* and simply used to describe situations of simultaneous speech. However, the term "interruption" is an *interpretative* category and identifies the result of a violent barge into the turn with the intention to takeover the other's right to speak. This is also stated clearly by Tannen (1994), when she writes that "affixing this label accuses a speaker of violating another speaker's right to the floor, of being a conversational bully".

TURN-TAKING
EFFECTS OF
INTERRUPTIONS

According to Edelsky (1981), a turn is always uttered with the "intention to convey a message that is both referential and functional", that means that an utterance clearly refers to something said earlier in the conversation. In contrast, verbal feedbacks, such as "mhm" or "yeah", are frequent and natural characteristic of conversations and joint activities but, by no means, a violation of turn-taking rules (Shriberg *et al.*, 2001; Oviatt *et al.*, 2015). In literature they are referred to as *encouragers* (Edelsky, 1981), *back-channels* (Yngve, 1970; Duncan, 1972; Sacks *et al.*, 1974; Allwood *et al.*, 1993), *listener responses* (Dittmann and Llewellyn, 1967), *minimal responses* (Orestrom, 1983; Zimmerman and West, 1975) or *accompaniment signals* (Kendon, 1967). They have been categorized into *supports* (e.g. "sure" or "right"), *exclamations* (e.g. "oh" or "hell"), and *exclamatory questions* (e.g. "what?" or "really?"), or are called *completions*, when a listener completes a speaker's sentence, *restatement* when a listener rephrases a speaker's statement in his own words (Orestrom, 1983), or *cooperative overlaps* when a listener uses them to signal enthusiastic listenership and high involvement (Tannen, 1984).

FEEDBACK
EFFECTS OF
OVERLAPS

Even if these feedback behaviors are used for very different reasons and the eventually resulting overlaps may considerably vary in length, they do not constitute claims for the turn (Duncan, 1972) and are commonly not considered to be interruptions (Schegloff, 1968). They can rather be regarded as a kind of positive reinforcement for continuing talk (Schegloff, 1968; Fishman, 1997), because they are often used by the listener to signal that he is not interested to take the turn in the conversation but to show continuing understanding, agreement, interest, engagement, and co-participation in topic development (Yngve, 1970; Allwood *et al.*, 1993) without requesting the turn. They can reflect empathy, enthusiasm, and indignation (Stenstrom, 1994) but can sometimes also signal negative attitudes, such as also a lack of interest, indifference, impatience, and non-support (Zimmerman and West, 1975).

2.3.3 Attitudes and Relationships

While *intrusive interruptions* demonstrate disagreement and can force turn- or topic changes, *cooperative interruptions* support the conversation by expressing involvement, solidarity (Tannen, 1994, 2012), affiliation, engagement, and cooperation (Li, 2001; Li *et al.*, 2004) and *agreement interruptions* signal compliance, understanding, interest, and enthusiasm (Kennedy

ATTITUDES
& GROUNDING

and Camden, 1983a). Other *clarification interruptions* help the partners to repair the common ground for further communication (Clark and Brennan, 1991), for example, when the listener is unclear about a piece of information and interrupts the speaker to request a clarification (Kennedy and Camden, 1983a). Finally, elevated simultaneous speech can also be associated with improved performance in the sense that the participants more actively mediate and refine their own thoughts (Shriberg *et al.*, 2001). It can show a higher activity and domain expertise during collaborative problem solving, when solutions are most actively generated, discussed, and refined (Oviatt *et al.*, 2015). In this, it has been observed, that the contents of interruptions are highly role-dependent, for example, experts or instructors use interruptions in an authoritative and meta-regulatory way whereas the non-experts' interruptions often function to ask questions and to apologize for making errors (Oviatt *et al.*, 2015).

DOMINANCE
& STATUS

Overlaps and interruptions are also regarded as indication for domination and higher status. For example, Octigan and Niederman (1979) state that “an interruption or overlap is taken as a violation and a sign of conversational dominance”. West and Zimmerman (1983) say, that interruption is “a device for exercising power and control in conversation” because it involves “violations of speakers’ turns at talk”. Interruptions are considered as acts of dominance (Karakowsky *et al.*, 2004; Youngquist, 2009) since they are consciously or unconsciously used to reduce another’s role as communicator (Kennedy and Camden, 1983b,a) and control the conversational topic. People with a high social status tend to interrupt more frequently and thus often gain more attention and access to the speaker role at the expense of their interaction partners with a lower status (Smith-Lovin and Brody, 1989). However, also contradictory observations have been made, for example, in conversations in which teachers, usually having a higher status than pupils, were interrupted more frequently by their students (Beattie, 1980).

SEX, GENDER
& POWER

Finally, some research found that usually men try to interrupt their interaction partners more frequent than women (Zimmerman and West, 1975; West and Zimmerman, 1983; Tannen, 1984) and identified different patterns of interruptions between same and mixed-gender interaction partners (Zimmerman and West, 1975). While the distribution of interruptions was rather equally divided during conversations between participants with the same gender, in interactions with mixed genders, nearly all interruptions of women’s speech was made by men. This asymmetry has often been attributed to an imbalance of power and status between men and women in most western societies in which men are more likely than women to assume they are authorized to barge into others speech and grab the conversational floor (West and Zimmerman, 1983; Zimmerman and West, 1975). While these results have been replicated, others found no differences (Dindia, 1987), or even that women interrupt more frequently than men (Kennedy and Camden, 1983b). Women at least try to achieve more overlapping speech when conversing with interaction partners that seem to be talkative, attentive, cooperative, or emotionally mature, than with those the are silent, introspective, aloof, or critical (Beattie, 1981a).

2.4 Summary and Conclusion

In this chapter, I reviewed the literature from social and behavioral sciences to provide an understanding of the two, rather abstract and not yet thoroughly comprehended, interactional phenomena, and the underlying behavioral functions and processes, that are in the focus of this thesis. In Section 2.1, I provided the theory and definitions on interpersonal coordination and grounding and discussed their possible interrelations and synergistic effects. Afterwards, in Sections 2.2 and 2.3, I explained how different gaze behaviors, voice overlaps, and interruption attempts, which are representative behavioral aspects that are prominent in all situated social interactions, have an influence on interpersonal coordination and grounding. They have been illuminated with respect to their social outcome and their social and regulatory role for conversational mechanisms and behavioral functions that contribute to interpersonal coordination and grounding. Among those are, for example, attention following, intention prediction, multi-modal disambiguation, back-channel eliciting, mental and emotional displays, intimacy regulation, turn management, and interruption handling.

What should be remembered after this chapter is that gaze mechanisms and voice activity overlaps play various roles in different behavioral functions that contribute to interpersonal coordination and grounding. Interpersonal coordination includes the mutual entrainment of interactional rhythm and tempo, the tight reciprocal meshing of behaviors as well as the similarity of behaviors in time, called interactional synchrony, and form, referred to as behavior matching. Gaze and voice behaviors contribute to interpersonal coordination, for example, through the production and recognition of turn-taking actions, the detection of interruption attempts, or the facial mimicry of emotional displays during mutual gaze. Grounding refers to the interaction partners' constant collaborative effort to establish, maintain, and repair of the common conversational and perceptual ground. Behavioral functions that contribute to grounding, and in which gaze plays essential roles, are, for example, attention following, intention prediction, multi-modal disambiguation, and the eliciting of back-channel signals.

PART II

CHALLENGES AND RELATED WORK

*“People who wait for the perfect time to act
never take action.”*

MIKE MAHLER

CHAPTER 3

CHALLENGES — MODELING TASKS, REQUIREMENTS AND SOLUTIONS

In Chapter 1, I motivated and illustrated my research objective, which is the design of a behavior and interaction modeling framework for social agents that enables the control and coordination of the many behavioral functions and processes that contribute to interpersonal coordination and grounding. In Chapter 2, I presented those behavioral aspects that are in focus of this thesis, in particular the roles of gaze behavior and speech activity for the aforementioned behavioral functions. However, the definitions of the entailed challenges have been somewhat descriptive, being fairly vague and intuitive descriptions of the hurdles that an author encounters when modeling the interactive behavior of social agents. They fail to bring us closer to the identification of the core concepts and features of the proposed modeling approach. It is therefore necessary to move towards a more detailed description of the difficulties and an operational definition of challenges that are tackled in this thesis.

The rather hard-to-pin-down challenge of modeling interpersonal coordination and grounding capabilities for social agents can be refined into *modeling tasks* and task-specific *modeling requirements* for the used modeling framework. In this chapter, I address these tasks and requirements from a language engineering perspective, presenting a task-based categorization of requirements and a mapping between tasks, requirements, solution concepts, and proposed formalisms. I briefly explain how the chosen modeling concepts meet the individual modeling tasks and requirements. Most of the requirements are illustrated based on the gaze and speech behaviors that can be observed in the introductory scenario in Section 1.2.

In the remainder of this chapter, in Section 3.1, I briefly introduce the tasks and requirements as well as the solution concepts and languages of the proposed modeling approach. In Sections 3.2 to 3.4, I discuss the individual modeling tasks and the task-specific requirements in more detail. These tasks and requirements are intended to serve as a basis for the comparison with relevant related work on multi-modal behavior and interaction modeling in Chapter 4. The concepts and languages of the proposed modeling approach, that are only briefly addressed in this chapter, will be explained in more detail by the approach design in Chapter 5 and illustrated in Chapter 6 based on a realistic exemplary application.

3.1 Subtasks, Requirements and Solutions

The challenges that an author is faced with when modeling the interactive behavior of social agents can be classified into three modeling tasks. These are, first, the close coordination of the social agents' behavioral aspects as well as the interaction and dialog flow, second, the processing and reasonable interpretation of user input and context events, and finally, the creation of credible and expressive behavior and dialog content. Each of these modeling tasks is characterized by a number of task-specific requirements that must be met by the behavior and interaction modeling approach developed in this thesis. The design of appropriate modeling concepts that allow effectively and yet intuitively meeting these requirements is the basic prerequisite for creating fully-fledged computational behavior and interaction models of social agents using this approach. The modeling requirements identified in this section will serve as basis and guideline for the design of suitable and expressive modeling concepts in this thesis. A categorization with the three modeling tasks and their task-specific requirements as well as the mapping of these tasks and requirements to the corresponding modeling concepts and languages can be found in Table 3.1.1.

Table 3.1.1: An overview of the modeling tasks, requirements, solution concepts and languages.

Modeling Task	Modeling Requirement	Modeling Concept	Modeling Language
<i>Coordinating Functions & Processes</i> (Section 3.2)	<i>Incremental & Reciprocal Meshing</i> (Section 3.2.1)	<i>State-Chart Variant</i>	<i>Behavior Flow State-Charts & Glue Language</i> (Section 5.4)
	<i>Parallel & Hierarchical Structuring</i> (Section 3.2.2)	<i>Parallel Decomposition</i>	
		<i>Hierarchical Refinement</i>	
	<i>Interruption & Coherent Resumption</i> (Section 3.2.3)	<i>Interruption Policies</i>	
<i>Interaction History</i>			
<i>Integrating Input & Context Events</i> (Section 3.3)	<i>Uniform Knowledge Representation</i> (Section 3.3.1)	<i>Feature Structures</i>	<i>Behavior Flow Query Language</i> (Section 5.3)
	<i>Well-Organized Working Memory</i> (Section 3.3.2)	<i>Logic Fact Base & Event History</i>	
	<i>Multi-Modal Fusion & Reasoning</i> (Section 3.3.3)	<i>Logic Calculus</i>	
<i>Creating Behavior & Dialog Content</i> (Section 3.4)	<i>Versatile Composition of Behavior</i> (Section 3.4.1)	<i>Behavioral Activities</i>	<i>Behavior Flow Script Language</i> (Section 5.2)
	<i>Flexible Integration of Knowledge</i> (Section 3.4.2)	<i>Placeholder Variables & Inline Value Insertion</i>	
	<i>Automatic Variability of Behavior</i> (Section 3.4.3)	<i>Scene Grouping & Blacklisting</i>	

The following listing briefly summarizes the modeling tasks, requirements and solutions depicted in Table 3.1.1. In Sections 3.2 to 3.4 I explain and illustrate the task-specific, technical and conceptual modeling requirements in more detail. An important meta-requirement pervading all tasks and requirements is the *practicability* of the chosen modeling concepts for rapid prototyping, even by non-experts. Besides providing sufficient *expressiveness*, they should efficiently reduce the modeling effort and complexity while improving the maintainability and reusability of the behavior and interaction models.

⚙️ **Coordinating Functions & Processes**

Human interaction is characterized by the continuous coordination and multi-directional interplay of the interaction partners' behaviors and actions. Modeling this mutual interaction requires the *incremental and reciprocal meshing* of input processing, knowledge reasoning and behavior generation. Synchronizing the underlying simultaneous and nested behavioral and computational processes and layers requires the *parallel and hierarchical structuring* of a social agent's behavior and interaction model. Finally, the quickly changing prioritization and seamless transitions between behavioral functions as well as the consistent reconstruction of conversations and behaviors after suspensions require the immediate *interruption and coherent resumption* of these processes.

– **Incremental & Reciprocal Meshing**

The continuous interpretation of behaviors and generation of behavioral feedback, the early prediction of intentions and interests, the quick detection of misunderstandings and recovery from errors as well as the constant recognition of complex multi-modal behavioral patterns, requires a fine-grained, step-wise interleaving of behavior recognition, knowledge reasoning and behavior generation based on probabilistic, semantic, and temporal constraints. The proposed modeling approach in this thesis enables this incremental processing and reciprocal meshing with a special *state-chart* variant (Harel, 1987; Harel and Politi, 1998), called *Behavior Flow State-Charts (BFSCs)*.

– **Parallel & Hierarchical Structuring**

The coordination of behavior and interaction requires the proper synchronization and tight interleaving of multiple concurrent, reciprocally intertwined, behavioral and computational processes. These processes implement various behavioral aspects, such as modalities, functions, and operations on different, nested levels that are responsible for input processing, multi-modal integration, pattern recognition, decision-making, behavior control and dialog flow management. This multi-threaded and hierarchical structuring is tackled with the *parallel decomposition* and *hierarchical refinement* of a *BFSC* that implements a social agent's behavior and interaction model.

– **Interruption & Coherent Resumption**

Immediate behavioral responses to unforeseen events and the fast adaptation to changed behavioral goals require the quick prioritization and interruption of behaviors. Thus, state transitions of *BFSCs* have *interruption policies* which allow the immediate, preemptive interruption of a state's execution and the direct abortion of all nested behav-

ioral and computational processes. *BFSC*'s implement an *interaction history* consisting of an automatically maintained *history memory* that can be accessed to use the collected information when modeling adequate reopening, reconstruction, and recapitulation strategies for the coherent resumption of interrupted behaviors and dialogs.

✂ *Integrating Input & Context Events*

The robust understanding of multi-modal behavior comprises the integration of information distributed via various communication modalities and context knowledge. The quite irregular occurrence and heterogeneous content of the corresponding input and context events calls for a *uniform knowledge representation* format. Their modality-specific processing delays and effect times necessitate the maintenance of a *well-organized working memory* to store them for modality-specific time periods in their original chronological order. Finally, their interpretation and combination requires a *multi-modal fusion and reasoning* calculus that allows the integration of the contained information and the involvement of context knowledge based on semantic, temporal, and quantification constraints.

– *Uniform Knowledge Representation*

The fusion of information from multiple modalities requires to cope with the heterogeneity and irregularity of the individual modalities' events as well as the integration of arbitrary new modalities and sensor devices. The common processing of these events and context changes calls for a universal representation of symbolic and semantic data on various abstraction levels and processing stages that is carried by the different kinds of input events and knowledge facts. The proposed modeling approach tackles this requirement with *feature structures* (Kasper and Rounds, 1986; Carpenter, 1992) that are encoded as list-based terms in *PROLOG* (Clocksin and Mellish, 1981).

– *Well-Organized Working Memory*

The varying processing delays and effect times of the individual modalities' events requires to preserve them in their real chronological order for modality-specific persistence times in order to make them available when they are needed for multi-modal fusion. In addition, domain and context knowledge might be too complex to be represented with the primitively typed variables of a *BFSC*. For these reasons, it is necessary to maintain a working memory including a well-formed multi-modal *event history*. The approach in this thesis therefore uses a *PROLOG fact base* which is managed with dynamic predicates of the domain-specific *Behavior Flow Query Language (BFQL)*.

– *Multi-Modal Fusion & Reasoning*

Reasoning on the aforementioned knowledge base, and, in particular the multi-modal fusion of events in the included event history, requires the evaluation of a variety of semantic, temporal, and quantification constraints (Allen, 1983; Oviatt *et al.*, 1997; Mehlmann and André, 2012). In addition, a sufficiently fast and real-time capable inference on this event history requires a proper size and content management. The *BFQL* meets these requirements with a collection of prepublished, built-in *PROLOG*

facts, rules, logic and dynamic predicates that together form the *multi-modal event logic* and *garbage collection* mechanisms of this domain-specific logic calculus.

Creating Behavior & Dialog Content

Human social behavior includes a versatile repertoire of individual behaviors, complex behavioral patterns and multi-modal utterances. Thus, the creation of a social agent's behavior and dialog calls for a very *versatile composition of behavior*. Furthermore, humans use their domain and context knowledge for planning credible and well-informed dialog contributions and actions. To make a social agent a competent conversation partner therefore requires the *flexible integration of knowledge* into his utterances, behaviors, and actions. Finally, humans show a natural variation of behavior and support the grounding process by rephrasing and reformulating statements. Equipping a social agent with this ability requires that the modeling approach allows the *automatic variability of behavior*.

– **Versatile Composition of Behavior**

Credible and expressive interactive behavior requires the creation and coordination of versatile behaviors and actions. Among those are verbal contributions and nonverbal cues, such as gestures, postures, facial expressions, head-, eye-, and gaze movements but also application-specific actions. These individual behaviors and actions must be combinable in various ways, to complex behavioral patterns, and multi-modal utterances in which co-verbal behaviors are aligned with spoken words. The proposed modeling approach allows the flexible specification of such behavior compositions called *behavioral activities* written in the *Behavior Flow Script Language (BFSL)*.

– **Flexible Integration of Knowledge**

Conducting a well-informed dialog requires to integrate contents that have been inferred from domain and context knowledge into nonverbal behaviors and multi-modal utterances. For example, the correct referencing of objects by means of their location or attributes requires the integration of position coordinates for pointing gestures and directed gaze as well as verbal feature descriptions into prepared sentence structures. The proposed modeling approach meets this requirement with the *inline insertion* of values into the specifications of behavioral activities as well as the substitution of *placeholder variables* when playing back parameterized *scene activity templates*.

– **Automatic Variability of Behavior**

Grounding content in natural conversations involves to rephrase or reformulate specific dialog contents or whole multi-modal utterances in order to resolve misunderstandings and recover the conversational ground after disruptions. Furthermore, the creation of credible and competent behavior requires the natural variation of behaviors and dialog contents as well as the avoidance of literal repetitions that would otherwise make the speaker appear less believable. The proposed modeling approach tackles this requirement with the possibility to automatically vary whole behavior blocks by a aggregating multiple alternatives for a scene activity to a *scene group* and automatically selecting individual alternatives at runtime based on a *blacklisting strategy*.

3.2 Coordinating Functions and Processes

As mentioned in Section 3.1 and outlined in Table 3.1.1, the first essential modeling task is the close coordination of behavioral aspects, this means the proper prioritization, synchronization, and interleaving of the many incremental, reciprocal, concurrent, and hierarchical, computational and behavioral processes. The first modeling requirement is the *incremental and reciprocal meshing* of input processing, knowledge reasoning and behavior generation. This is a prerequisite for the continuous behavioral feedback to the partners' behaviors, prediction and proactive response to their intentions and interests as well as the fine-grained, step-wise recognition of multi-modal, multi-directional behavioral patterns. The second requirement is the *parallel and hierarchical structuring* of the behavior and interaction model. This means its division into concurrent and nested processes and layers that implement different behavioral or computational functions through their interplay. These processes must be coordinated using appropriate synchronization and inter-process communication mechanisms. This also allows a distributed development which efficiently reduces the modeling effort and complexity while improving the maintainability, extensibility and reusability of a model. Finally, the third requirement is the immediate *interruption and coherent resumption* of behaviors and interaction phases. This is necessary for direct responses to the others' behaviors, environmental distractions, or one's own changing behavioral goals and priorities as well as the recapitulation and consistent reopening of suspended behaviors and interaction phases.

3.2.1 Incremental and Reciprocal Meshing

CONTINUOUS
BEHAVIORAL
FEEDBACK

As mentioned in Chapter 2, the participants of joint activities constantly monitor their partners' actions and behaviors and continuously determine appropriate reactions and adjustments of their own behavior in response (Clark and Krych, 2004; Clark, 2005; Brennan *et al.*, 2008). This incrementally contributes to a better understanding of their partners' behaviors and a more precise picture of their potential interests and intentions (Baron-Cohen *et al.*, 2001; Meltzoff and Brooks, 2001; Tomasello, 1995). Vice-versa, they continuously provide feedback themselves, for example, by producing back-channels to signal agreement or understanding (Kendon, 1967; Yngve, 1970; Allwood *et al.*, 1993; Bavelas *et al.*, 2002), and, constantly reveal their point of reference (Sebanz *et al.*, 2006; Mundy and Newell, 2007) in order to reduce the collaborative effort (Clark and Brennan, 1991; Clark and Krych, 2004) for sharing a common perceptual ground (Clark, 2005). Thus, they make the interaction fluid (Hough and Schlangen, 2016) and can detect misconceptions or recover from errors at an early stage (Brennan *et al.*, 2008; Sebanz and Knoblich, 2009; Huang *et al.*, 2015).

There can be found many examples for the tight incremental meshing of behavior recognition, knowledge reasoning, and the continuous generation of behavioral feedback in the introductory scenario in Section 1.2. For instance, in scene ①, Marley's gaze is permanently wandering across the surface table while, in response, Charly is constantly following her gaze to those photos that are particularly attracting her attention. Charly's behavior in this scene requires the steady, seamless interleaving of the observation of Marley's eye and head move-



ments, the proper identification of her gaze targets, and, at the same time, following of her gaze shifts to the photos on the surface table in order to track her focus of attention and share the same perceptual ground. Another kind of incremental processing, the intermeshing of behavior interpretation and knowledge reasoning is, for example, necessary for the proactive presentation of information about specific photos that Marley's might be interested in, as described in scene ② of the introductory scenario. When a specific photo is catching Marley's attention over a certain period of time, then Charly assumes that she is particularly interested in this photo and either asks if she wishes more information or anticipates her wish and provides this information, even without being asked before. Therefore, it is necessary that he constantly infers information about the content which is depicted on Marley's currently focused photo from its domain knowledge and that he plans appropriate formulations to present this information.

Chapter 2 has also described that the participants of joint activities use multi-modal behavioral patterns to produce particular social and regulatory signals. These behavioral patterns are composed of verbal utterances, and well-aligned accompanying nonverbal behaviors and co-verbal actions. They are not necessarily produced by a single participant, but are more often bi- or multi-directional, which means they result from the step-wise, reciprocal interleaving of behaviors and actions produced by several involved interaction partners. Often an individual behavior's contribution is not perfectly unambiguous when it is first recognized, but reveals its meaning after subsequent behaviors have been awaited and considered for the combination with it to a behavioral pattern based on semantic and temporal relations (Oviatt *et al.*, 1997). Typical examples for such multi-modal behavioral patterns have been described in Chapter 2, such as the various turn-taking actions (Nielsen, 1962; Duncan, 1972, 1974; Goodwin, 1980, 1981) and conflicts caused by voice activity overlaps (Schegloff, 2001; Goodwin, 1981) as well as the different feedback eliciting cues (Kendon, 1967; Allwood *et al.*, 1993; Bavelas *et al.*, 2002) and connection events (Rich *et al.*, 2010; Holroyd *et al.*, 2011).

BEHAVIOR
PATTERN
RECOGNITION

The introductory scenario in Section 1.2 is full of bi-directional behavioral patterns that need to be recognized incrementally. One of them, which is used for coordinating the exchange of the participant roles, can be found in the scenes ⑧ and ⑨. In this situation, the end of Marley's utterance "I miss these old times!" in scene ⑧ serves as an indication for Charly that she might start a turn-taking action, however, Marley's behavior is still ambiguous at this point in time because it could be the first step of a hold, yield or assign action. Since Marley avoids to look at Charly during her utterance and also in the subsequent moments of scene ⑧, Charly interprets Marley's behavior as turn-hold pattern, assumes that she wants to keep the floor further on and, for that reason, continues following her attention for the moment. During the next situation in scene ⑨, he awaits the point in time at which Marley has first finished her utterance "Get what?" and then both have additionally established mutual facial gaze shortly afterwards, which represents a gaze connection event that completes the bi-directional pattern for the assignment of the turn. This is the definitive proof that Marley is offering the floor to Charly who confidently accepts this offer in the second part of scene ⑨ by asking "Would you like some tea?".



The above examples clearly demonstrate that natural social interactions and joint activities are characterized by the fine-grained, reciprocal meshing of their participants' behaviors and actions. This allows them to generate and recognize multi-modal and multi-directional behavioral patterns and to continuously provide and recognize behavioral responses and thus predict intentions and interests, detect misunderstandings, and recover from errors at an early stage. In order to equip a social agent with these capabilities, the behavior and interaction modeling approach must master the incremental processing and reciprocal interleaving of input processing, knowledge reasoning and behavior generation. As later explained in Chapter 5, the proposed modeling approach meets this requirement by relying on a specifically designed state-chart variant (Harel, 1987; von der Beeck, 1994; Harel and Naamad, 1996; Harel and Politi, 1998; Harel et al., 1990; Harel and Kugler, 2004), called *Behavior Flow State-Charts (BFSCs)*. This *state-transition-based* modeling approach with *BFSCs* allows the fine-grained interleaving of computation steps for input processing, knowledge reasoning and behavior generation based on conditional, timed, and probabilistic strategies.

3.2.2 Parallel and Hierarchical Structuring

As described in Chapter 2, the participants' behavior in natural social interactions includes various parallel and hierarchical aspects of behavior which are all together managed and coordinated at the same time. People simultaneously use multiple behavioral modalities, such as speech, gestures, postures, facial expressions, and gaze cues to exchange information and social cues (Jaimes and Sebe, 2007; Oviatt, 2012). The individual behaviors in these modalities are carefully aligned with each other to create different multi-modal behavioral patterns, which, for their part, contribute to one or more specific behavioral functions, such as following the partners' attention (Kendon, 1967; Mundy and Newell, 2007), predicting their next actions (Sebanz and Knoblich, 2009; Huang et al., 2015), disambiguating their multi-modal references (Oviatt, 1999; Kaiser et al., 2003; Kaur et al., 2003; Hanna and Brennan, 2007; Oviatt, 2012), or making decisions concerning the participant role regulation (Sacks et al., 1974; Schegloff, 2000) and turn-conflict handling (Bennett, 1981; Tannen, 1994; Schegloff, 2001). Finally, other behavioral factors, such as cognitive processes and mental states, emotional conditions and moods, personality traits and physical status as well as social dimensions, such as dominance, engagement, and politeness simultaneously find their expression in the observable behavior of the interaction partners (Argyle, 1972; Mehrabian, 1972; Argyle, 1975; Picard, 1997; Knapp et al., 2014).

From an author's perspective, the aforementioned behavioral aspects can be understood as the result of a complex interplay of multiple concurrent and nested behavioral and computational processes on different behavioral levels and processing stages. This suggests to structure a social agent's behavior and interaction model such that it may be refined into hierarchical layers which are then decomposed into parallel components and vice versa. These components then have responsibilities roughly corresponding to the behavioral aspects or more individual tasks contributing to a specific aspect, such as the recognition of behavioral patterns or the generation of role- and context-dependent behavior. They must then

be closely and reasonably synchronized with each other such that they together contribute with their complex but coordinated interplay to the observable behavior of the social agent.

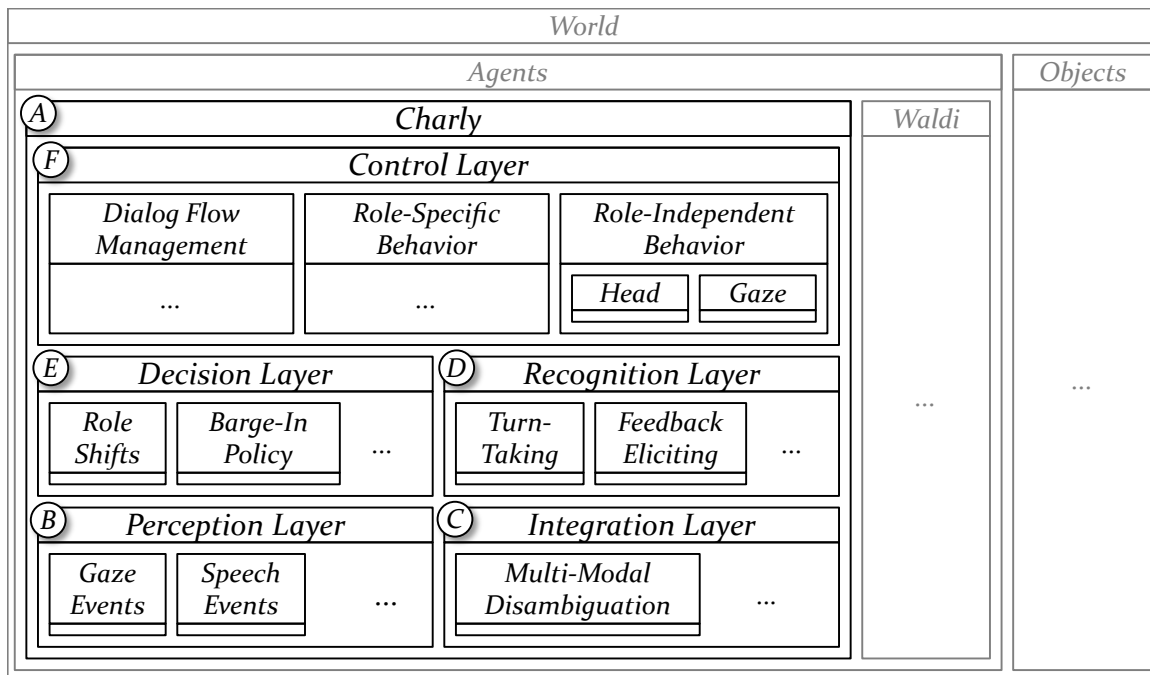


Figure 3.2.1: An illustration of the parallel and hierarchical decomposition of the behavior model.

Figure 3.2.1 outlines an exemplary hierarchical and parallel structuring of the interaction model for the scenario in Section 1.2. The model represents objects and agents in the physical environment as individual nested and parallel processes and layers. The part containing Charly's behavior and interaction model is depicted in more detail (Figure 3.2.1 (A)). Processes on the *perception layer* (Figure 3.2.1 (B)) are preprocessing Marley's inputs and context changes, such as the ringing kettle in scene ⑦. They are responsible for populating Charly's knowledge with the contained information and its propagation to higher levels of the model. An example is the process computing Marley's gaze shifts to objects, persons, or events based on the gaze target distributions provided by the eye-tracking glasses. Processes on the *integration layer* (Figure 3.2.1 (C)) are, amongst other things, responsible for the multi-modal fusion of Marley's inputs based on semantic, temporal and quantification constraints. An example is the disambiguation of the deictic referring expression "this" in Marley's utterance "Where is this beach?" with information from her gaze shifts in scene ③. Processes on the *recognition layer* (Figure 3.2.1 (D)) recognize multi-modal and bi-directional behavioral patterns, such as Marley's coordinated use of speech and gaze for the production of the turn-taking actions observable in scene ⑧ and ⑨, or the feedback eliciting cues in scene ⑩. On a superordinate *decision layer* (Figure 3.2.1 (E)), information from lower and higher layers is used to make decisions concerning the behavior and interaction management, such as the assignment of participant roles in reaction to turn-taking actions or conflicts. A process on this layer might, for example, contextually decide if Marley's attempt to take the turn may interrupt Charly and allow Marley to occupy the speaker floor, as happening in scene



⑧. Finally, several processes on the behavior and dialog *control layer* (Figure 3.2.1 ⑥) manage reactive aspects of Charly’s behavior, that means physiological reactions and ideomotor nonverbal behaviors, like Charly’s gaze following behavior in scene ① or his cognitive and emotional mimicry in scenes ⑤ and ⑥. Other processes on this layer control deliberative aspects of behavior, such as the dialog flow management and the inference of knowledge about the photos when producing answers to Marley’s questions, like those in scene ③.

The numerous behavioral and computational processes in a model like that depicted in Figure 3.2.1 are only in the rarest cases operating completely independent from each other. In contrast, credible and expressive behavior of an agent can only be created through the closely coordinated interplay between these processes. Therefore, it is necessary to provide appropriate mechanisms for synchronization and inter-process communication that allow an author to exchange information between and properly interleave the execution of concurrent processes (Lampert, 1986a,b). The asynchronous exchange of events is a non-blocking mechanisms that can be used to pass control signals and transfer information between individual parallel processes in different parts of a model. The sending process proceeds with its task and must not wait until the receiving process has consumed the event and has eventually acknowledged the delivery with a separate event. Another method to exchange information is the mutual exclusive access to variables and facts in a shared memory or configuration space that are more likely used to represent more persistent information which may be observed by multiple processes. The actual read and write access to the shared memory or the assertion and retraction operation on a shared knowledge base are blocking since they represent critical sections in order to ensure the consistency of the memory or knowledge.

Examples for the coordination and interleaving of concurrent processes can be found in various situations of the introductory scenario in Section 1.2. For example, the attention following behavior, shown in scene ①, can be realized by the step-wise interleaving of a process on the perception layer that recognizes Marley’s gaze shifts, and a process on the control layer that causes Charly to follow these gaze shifts. The communication between these two processes can be realized via both events or variables. However, the process that is monitoring Marley’s voice activity on the perception layer is more likely to represent her speaking state in a shared variable such that it can be read by multiple processes on the recognition layer, like those detecting speech overlaps and turn-taking actions, or conflicts, coming to effect in scenes ⑧ and ⑨, or the one that is recognizing feedback eliciting cues, as observed in scene ⑩. The resulting signals, in turn, would again be represented as events and propagated to the processes on the decision and control layer.

The above examples show that human behavior and social interactions exhibit a highly parallel nature. Humans closely coordinate multiple parallel and highly interwoven aspects of behavior, such as behavioral functions, modalities, levels, and underlying processes. Equipping a social agent with this capability requires a modeling approach that can control concurrent behavioral and computational processes and offers appropriate inter-process communication mechanisms for their synchronization and information exchange. As explained in Chapter 5, the proposed modeling approach meets this requirement with the *hierarchical*



refinement and *parallel decomposition* of the behavior and interaction model (Harel, 1987; Harel and Politi, 1998). Nested *BFSCs* can be used for the context-dependent modeling of input processing and behavior generation. Parallel *BFSCs* can be used to model individual behavioral aspects in separate processes, and properly synchronize them using a *shared memory* or asynchronously exchanging *signal events*. Following the *divide and conquer* principle, this distributed approach significantly reduces the modeling effort and complexity while improving scalability, reusability, and extensibility of a model.

3.2.3 Interruption and Coherent Resumption

As demonstrated in Chapter 2, spontaneous changes of the participants' behaviors due to changing behavioral goals and priorities or suddenly distracting events in the physical environment are hardly foreseeable by their interaction partners. However, humans compensate for the lack of foresight in these situations, to some extent, with their ability to quickly detect and interpret such behavioral and contextual changes and to immediately suspend and adapt their own behavior in response. This enables them to keep up with their partners' interaction tempo and rhythm (Davis, 1982; Hayes and Cobb, 1982), improves their interactional synchrony and simultaneity (Bernieri and Rosenthal, 1991), and therefore contributes to their interpersonal coordination (Bernieri and Rosenthal, 1991; Richardson *et al.*, 2005; Delaherche *et al.*, 2012). Immediate behavioral responses allow smoothly following the others' perceptual ground and constantly sharing a common point of reference (Brennan, 1998; Sebanz *et al.*, 2006; Mundy and Newell, 2007). Instant self-interruptions and reactions to a partner's barge-in attempt, signaling misunderstanding, non-understanding, or misconceptions (Hirst *et al.*, 1994) helps to quickly recognize the need for clarifications and to speed up the reestablishment of the common conversational ground (Clark and Brennan, 1991; Clark, 1996).

IMMEDIATE
BEHAVIORAL
RESPONSES

An example that an immediate behavioral response is beneficial for a natural and credible behavior can be found in scene ⑦ of the introductory scenario in Section 1.2. The joint activity of Marley and Charly is suddenly disturbed by the ringing kettle which, in this case, represents an environmental distraction. Charly shows a prompt and very natural reaction by reflexively shifting his focus of attention to the kitchen in which the kettle is located. His behavior, however, causes a sudden change of the interactional tempo and rhythm, to which Marley, in turn, is adapting in a natural way because she is immediately entrained by Charly's reaction and follows his focus shift just as quickly. Charly's behavior also influences the grounding process because it might be unclear how to proceed with the common task after this distraction. As shown in the next scenes ⑧ and ⑨, both are, however, able to quickly renegotiate and reconstruct their primary conversational topic, which are the photos on the table, and find back to their previous interaction rhythm again.



Very similar to such a reaction and adaptation to external events, the participants of joint activities also interrupt and adjust their behaviors of their own accord, on the basis of revised behavioral goals and priorities. The current behavioral goal or function is, at least temporary, given up in favor of another objective that is considered to be more important

CHANGING
GOALS &
PRIORITIES

or appropriate in a particular situation because of greater benefit, relevance, or urgency. For example, an intervention, due the prediction of the partner's misunderstanding, can reduce the collaborative effort to maintain the common ground and, thus, ensure the efficiency and success of the collaborative task (Clark and Brennan, 1991; Clark and Krych, 2004; Clark, 2005). Other behavioral adjustments are important for interpersonal coordination, for example, when avoiding or quickly reacting to speech overlaps with the interaction partners that could cause turn-taking conflicts (Schegloff, 2000, 2001; Goodwin, 1980, 1981) in order to ensure a seamless exchange of the participant roles (Sacks *et al.*, 1974; Bernieri and Rosenthal, 1991).

The continuous adaptation of the behavior due to the changing priorities of the different functions can be observed all over the entire interaction in the introductory example scenario in Section 1.2. An example for the interruption and resumption of gaze behavior due to different functional priorities can be found in Charly's different gaze behaviors in the scenes ① and ⑦. In general, Charly performs an ordinary listener or bystander gaze behavior by dividing his visual attention between Marley and points or objects in the environment, such as for example the photos on the surface table or the objects in Marley's apartment. However, as may be observed in scene ①, it is more important for the grounding process, that he shares the same perceptual ground with Marley and therefore follows Marley's occasional gaze movements and pointing gestures to the photos on the surface table while being in the role of an attentive addressee. Nevertheless, a sudden and unexpected distraction, such as the ringing of the kettle, occurring in ⑦, in turn, is an urgent and natural reason to interrupt the gaze following behavior with a short glance of gaze to the source of the distraction, after which, however, the gaze following behavior is resumed again, as observed in scene ⑧.

COHERENT
RESUMPTION
OF BEHAVIOR

This example already demonstrates that suspended behaviors are frequently resumed at a later point during the interaction after they have regained priority again. In this, humans exploit different strategies for a coherent resumption of behaviors with the aim to reduce the collaborative effort for reestablishing the common ground. In some cases, as in the above described resumption of gaze following after a distraction, it is sufficient to simply pursue the previously aborted behaviors at the point of interruption. However, other behaviors or interaction phases require a more sophisticated reopening or recapitulation in order to coherently and seamlessly continue with them. For example, when interrupted by a third party or environmental event, the interaction partners are able to quickly suspend the original interaction, deal with the interruption, and afterwards coherently reconstruct and reinstate the primary interaction again (Bangerter *et al.*, 2010). In this, the conversation partners often recapitulate a part of what has been said and done during a short subdialog in order to reestablish the common ground (Gandhe and Traum, 2008) after a distraction or barge-in (Ferguson, 1977; Beattie, 1981a; Tannen, 1994; Li, 2001; Schegloff, 2001).

Examples that such coherent resumptions of previously interrupted behaviors are necessary for the plausibility of a social agent's behavior can be found in scenes ⑦ to ⑨ of the introductory scenario in Section 1.2. In scene ⑦, by asking "Marley, should I get a cup ...", Charly tries to direct the topic of the conversation on the ringing kettle, or, in better words, to the tea

that they planned to drink before they started the photo book application on the surface. Barging into Charly's utterance in scene ⑧ by saying "I miss these old times!", Marley immediately interrupts Charly's attempt to change the conversational topic and directs the dialog on the photo in front of her instead. Afterwards, in scene ⑨, she tries to reopen the interrupted topic again by asking "Get what?" and Charly recapitulates by paraphrasing his interrupted question, saying "Would you like some tea?". Finally, Marley refuses the offer to drink a tea by saying "No, thanks!" such that both agree upon the photos as the next conversational topic and have thus reestablished the common ground as the conversation continues.

The above examples show that, even though, some behaviors might appear unforeseen, the participants of social joint activities are able to immediately detect, interpret, quickly interrupt themselves, and seamlessly respond to their partners' behaviors or adjust their behavior to the changing priority of their own behavioral goals and functions. In addition, they use appropriate reopening strategies and recapitulation phases for the consistent reinstatement of previously interrupted behaviors and dialogs. In order to equip a social agent with these abilities, the modeling approach needs to master the immediate interruption of behavioral and computational processes and their coherent resumption. As demonstrated in Chapter 5, the proposed modeling approach meets these requirements by combining the *hierarchical refinement* of *BFSCs* with special *interruption policies* for transitions. Priorities between competing behavioral aspects can then be implemented by nesting behaviors with lower precedence into deeper levels in the hierarchy and using interruptive transitions that preemptively abort all subordinated behaviors on deeper levels. An automatically maintained *history mechanism* collects runtime information and built-in *history statements* of the *Behavior Flow Glue Language (BFGL)*, the textual expression language which is used with *BFSCs*, allow accessing this information to model coherent resumption strategies.

*INTERRUPTION
& COHERENT
RESUMPTION*

3.3 Integrating Input and Context Events

As addressed in Section 3.1 and depicted in Table 3.1.1, the second key modeling task is the proper integration of input and context events for the correct understanding of the partners' multi-modal behaviors and environmental changes. The first requirement is a *uniform knowledge representation* format for input events, domain, and context knowledge. This facilitates handling the heterogeneity and irregularity of observed events and thus ensures the compatibility and extensibility of the modeling approach. The second requirement is a *well-organized working memory* for the consistent management of dynamic knowledge including a well-formed, multi-modal event history to consider modality-specific processing delays and effect times at multi-modal fusion. A scalable garbage collection mechanism must regularly gather up outdated and irrelevant events to keep the inference sufficiently real-time capable. Finally, the third requirement is an expressive and practicable method for *multi-modal fusion and reasoning* on the working memory. This is needed for knowledge management, especially the integration of information distributed via events from the multiple modalities based on diverse semantic, temporal, and quantification constraints.

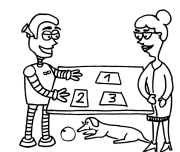
3.3.1 Uniform Knowledge Representation

The participants of social interactions exchange information using messages and signals in a multitude of modalities and communication channels, such as speech, gaze, gestures, postures, and facial expressions (Jaimes and Sebe, 2007; Oviatt, 2012). In physically situated, collaborative joint activities, such as the introductory scenario in Section 1.2, they additionally exchange information through the joint manipulation of objects on the shared workspace (Clark and Krych, 2004; Clark, 2005). In a human-agent interaction the agent’s modality-, sensor-, or device-specific recognition modules interpret this information and forward it in form of multi-modal events to the agents’ *multi-modal fusion engine* for further processing and multi-modal integration (Jaimes and Sebe, 2007; Lalanne et al., 2009; Dumas et al., 2009b). Structure and content of these events are rather heterogeneous, that means they can carry information on different processing stages and abstraction levels ranging from largely unprocessed and symbolic data, such as, for example, gaze coordinates, to completely interpreted and highly abstract semantic contents, such as, for example, dialog acts (Bunt et al., 2010; Bunt, 2011). The processing stage or abstraction level of the information can sometimes be, but is not necessarily, related to the frequency and regularity of the corresponding input events. Continuous behaviors and interactions usually produce more frequent events in rather regular intervals and with less abrupt changing contents. In contrast, discrete behaviors and actions result in more irregular and rare events whose contents can significantly differ.

Examples of input events with different frequency and regularity carrying heterogeneous information are found in various scenes of the introductory scenario in Section 1.2. For example, while Marley’s gaze is wandering over the table in scene ⑥, Charly could continuously follow her gaze by processing the raw eye gaze coordinates of the eye-tracker or step-wise follow her gaze fixations provided by an upstream preprocessing component that is averaging and down-sampling the raw gaze data. While the gaze coordinates are symbolic information produced each frame, the fixations have a lower frequency, and carry higher level semantic information denoting the photo that Marley was most probably looking at during the last frames. The still regular, but rarer, fixations are used by Charly when predicting Marley’s interest in a specific photo, as in scene ②, or disambiguating Marley’s deictic references, as in scene ③. In parallel, Marley’s spoken utterances are recorded by a *Voice Activity Detection* (VAD) and processed by an *Automatic Speech Recognition* (ASR) module before being interpreted by a *Natural Language Understanding* (NLU) component which produces dialog acts that represent information on an even more abstract and interpreted semantic level. Carried by speech events, they arrive at the fusion engine in irregular intervals and might then be fused with the regular stream of gaze fixations for multi-modal disambiguation (Meyer et al., 1998; Griffin and Bock, 2000; Griffin, 2001; Kaur et al., 2003).

The growing technical advancement in sensor technologies brings new sensor devices more and more into our daily life and makes them available and affordable for a large number of users. Such devices support features, such as, for example, marker-less full body motion sensing, mobile eye-tracking, and capturing of biological signals, for example, for affective

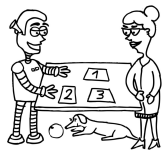
HETEROGENEOUS
& IRREGULAR
INFORMATION



TECHNICAL
PROGRESS &
COMPATIBILITY

computing (Picard, 1997). Social companion technologies and the instrumented intelligent environments, that they jointly occupy with humans, are more and more integrating such modern devices (Fong *et al.*, 2003; Leite *et al.*, 2013). This development poses new conceptual and technical challenges to the capabilities of multi-modal fusion engines. To keep pace, they may not be restricted to a specific application domain, an unchangeable sets or specific combinations of modalities and input devices, or mandatory or primary modalities that are indispensable for their underlying formalism to work. Instead, they must allow easily integrating novel devices and modalities in order to exploit the full potential of multi-modal fusion and thus promote the evolution towards natural interaction with social agents.

An example, how the interchangeability and flexible combination of modalities helps to exploit the potential of multi-modal communication can be found in scene ③ of the introductory scenario in Section 1.2. In this scene, Marley refers to a specific photo by saying “Tell me! Where is this beach?” while directing her gaze to this photo. In this case, she relies on Charly’s ability to disambiguate the referring deictic expression “this” with the information from her gaze behavior. However, she could as well have chosen other ways of referring to the photo on the shared workspace, for example, by using only directed gaze, a deictic pointing gestures, or a verbal referring expressions, but also any combination of these modalities. In order to understand all of Marley’s possible referring strategies, Charly’s fusion engine must be able to combine all these modalities without being completely reliant on any of them.



The above examples show, that social agents’ fusion engines must offer the full power of multi-modal fusion support across different levels of abstraction and processing stages (Hall and Llinas, 1997; Dasarathy, 1997; Sharma, 1998; Bosma and André, 2004; André *et al.*, 2014; André, 2014). They must use a generic and uniform knowledge representation format in order to process streams of frequent and regular events produced by continuous behaviors, such as eye gaze and object manipulations, as well as concurrently occurring, rare, and irregular events, for example, pointing gestures and spoken utterances, carrying higher-level symbolic and semantic information. As explained in Chapter 5, the proposed modeling approach tackles these challenges by representing input and context knowledge with *feature structures* (Kasper and Rounds, 1990; Carpenter, 1992; Pereira, 1993) encoded as list-based *PROLOG* facts which are managed using dynamic *PROLOG* predicates that are part of the domain-specific logic calculus, referred to as *Behavior Flow Query Language (BFQL)*. In this way, all modalities are treated equally and no restrictions for their combination possibilities are implemented, thus, efficiently facilitating the compatibility and extensibility of the modeling approach.

UNIFORM
KNOWLEDGE
REPRESENTATION

3.3.2 Well-Organized Working Memory

The computational effort from the detection of a raw input signal to its semantic interpretation causes varying, modality-specific processing delays. While these might hardly be noticeable in human interactions, due to the humans’ superior perception skills and cognitive capacity, they, however, represent a technical challenge for multi-modal fusion engines of social agents. Those might incorrectly interpret multi-modal inputs because they could receive an input event stream in a temporal order that does not correspond to the real chronological

DIFFERENT
PROCESSING
TIMES

sequence of the underlying behaviors (Bellik, 1997, 2001). The modality-specific recognizers must therefore equip all produced events with timestamps for their producing behaviors' occurrences and lifespans. The fusion engine must buffer them in an event pool to reconstruct their real order and compensate for the different and changing, modality-specific processing delays. A fusion calculus can then take account of the time that it took from the detection of a behavior, over its interpretation, until a corresponding event has been forwarded to the fusion engine (Portillo *et al.*, 2006; Dumas *et al.*, 2009a; Johnston, 2009).

**DIFFERENT
PERSISTENCE
TIMES**

In addition to varying processing times, individual behaviors and modalities also have different persistence times, in the sense that their effect on the understanding of multi-modal utterances and behavioral patterns can be felt for differently-sized time windows after their occurrence. Processing delay and effect times are not directly correlated but it can often be observed that events carrying highly abstract and interpreted information have usually both longer processing delays and effect times than those with less processed and more symbolic data because they have undergone more and costlier processing steps. For example, an utterance often transports persistent information, such that the corresponding dialog act affects the discourse history for a substantial period of time. In contrast, gaze movements and fixations represent rather transient and volatile information of the user's visual orientation and attention. In order to integrate these different types of event persistences, it is necessary to maintain a short-term memory of input events in which events are retained according to their potential, modality-specific persistence (Hoste *et al.*, 2011; Mehlmann and André, 2012).

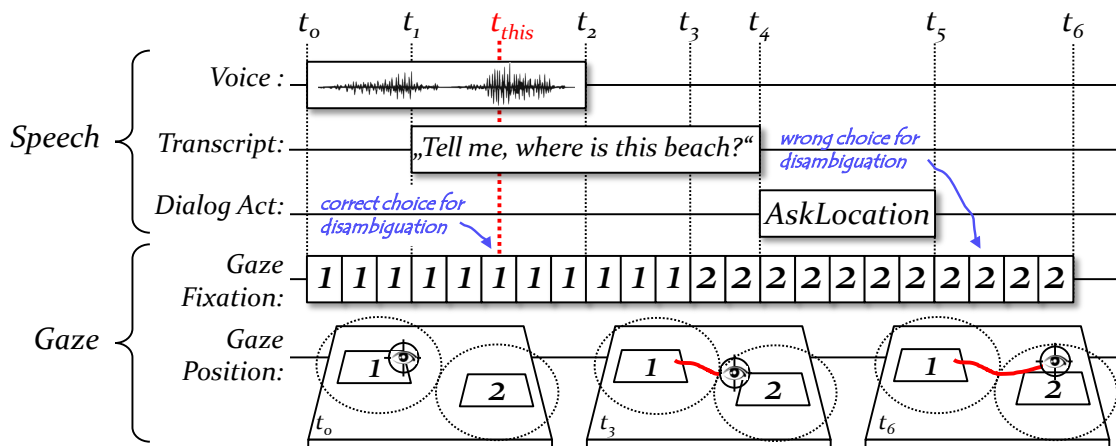


Figure 3.3.1: An illustration of the delays caused by the processing times of speech and gaze inputs.

Some technical delays that arise due to the processing of speech and gaze are illustrated in Figure 3.3.1 based on scene ③ of the introductory scenario in Section 1.2. Marley asks “Tell me! Where is this beach?” while looking at a photo on the surface table. The time line in Figure 3.3.1 depicts the duration of the processing steps of her verbal input and eye-gaze data. While Marley is speaking from t_0 to t_2 , Charly’s ASR module starts processing the voice stream at t_1 and has produced a transcript at t_4 . His NLU starts parsing the transcript at t_4 and returns a dialog act interpretation at t_5 . In parallel, Marley’s eye-gaze is moving on the table from the photo with number 1 to the one with number 2 while the eye-tracking module

regularly computes the most likely fixated photo. From t_0 to t_3 the photo with number 1 and from t_3 to t_6 the one with number 2 is determined as Marley's gaze target. Charly's fusion engine must disambiguate the underspecified dialog act after t_5 , by considering her gaze direction. Without storing events and looking at their timestamps, it would have to choose Marley's gaze fixations after t_5 which all refer to the wrong photo with number 2. However, the correct choice would be the photo with number 1 which Marley's looks at about 600 to 800 milliseconds before the deictic referring expression "this" (Meyer *et al.*, 1998; Griffin and Bock, 2000; Griffin, 2001; Kaur *et al.*, 2003) within her utterance.

The above example illustrates that incoming events must be handled based on their real chronological sequence in order to correctly interpret their combined meaning for multi-modal utterances and behavioral patterns. Due to different and changing, modality-specific processing times for the acquisition, recognition, and interpretation of the underlying behaviors, they must be equipped with timestamps and managed in a short-term memory. Outdated events must regularly be removed by a garbage collection to keep the inference mechanism real-time capable. As explained in Chapter 5, the proposed modeling approach tackles these requirements with a *well-formed event history* as part of a *PROLOG* fact base (Kowalski, 1974; Emden and Kowalski, 1976; Kowalski, 1979). The therein contained events must have all relevant timestamps and be totally ordered within the same modality (Allen, 1981, 1984; Allen and Hayes, 1990; Allen, 2013). Dynamic *BFQL* predicates are used to realize an age-based and modality-specific *garbage collection*, such that frequently received events carrying low-level symbolic data, such as those from gaze or touch behaviors, are retained for a shorter time than events with highly abstract semantic information, such as, for example, dialog acts.

WELL-
ORGANIZED
WORKING
MEMORY

3.3.3 Multi-Modal Fusion and Reasoning

A critical challenge in social interactions is the correct understanding of the partners' multi-modal utterances and behavioral patterns. Their robust interpretation and reduction of uncertainty can be achieved through the mutual disambiguation of the analysis results provided by each communication channel or modality (Oviatt, 1999; Kaiser *et al.*, 2003; Jaimes and Sebe, 2007; Oviatt, 2012; Mehlmann *et al.*, 2014a, 2016). This requires a reasonable method for the multi-modal fusion of partial semantic information distributed via events from multiple modalities. Besides the rather application-specific semantic relations, a social agent's fusion engine must consider temporal and ordering constraints to determine if individual contributions from different modalities are properly aligned and may be integrated based on their occurrence and duration. Quantitative temporal constraints allow the representation of temporal evolutions related to a given period of time or at a precise moment in time. Qualitative temporal relations, such as simultaneity, overlap, or inclusion are usually defined over time intervals (Allen, 1983; Allen and Ferguson, 1994). Ordering relations are, for example, used to determine the followers or ancestors of an event or perhaps the oldest or latest event of a set of events (Mehlmann and André, 2012; Mehlmann *et al.*, 2016).

TEMPORAL &
ORDERING
CONSTRAINTS

A user's gesture or utterance must occasionally be disambiguated with continuous, but often noisy and flawed information, such as gaze coordinates from an eye-tracker, emotion values from a prosody analysis, or physiological data from biosensors (Bosma and André, 2004; André, 2014). Then, it is helpful to inspect the temporal development of this data over a certain time window and select the most promising samples for disambiguation instead of just one or all of them. Such a conditional quantification can reduce uncertainties due to data loss, recognition errors, and outliers, and thus helps to more precisely resolve ambiguities (Prasov and Chai, 2008; Qu and Chai, 2008; Fang *et al.*, 2009; Prasov and Chai, 2010; Liu *et al.*, 2013).

CONDITIONAL
QUANTIFICATION
CONSTRAINTS

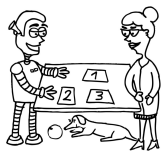
For example, a gaze recognizer based on eye-tracking and object recognition, as in the introductory scenario in Section 1.2, must cope with two main error sources. Technology-related false or irregular measurements can be due to different types of noise, sensor device and software errors, data losses caused by improper light conditions or viewing angles or imperfect parameter calibrations and transformation algorithms (Freeman *et al.*, 2007). Biology-related outliers or data losses are due to tremor and micro-saccades, eye jittering and blinking as well as the natural random offset due to fuzzy fovea dimensions between the vector of actual attentive gaze direction and eye optical axis (Špakov, 2011). Such a recognizer usually estimates the user's gaze target by comparing the gaze coordinates to the positions of the objects coming into question. It rates the objects in each frame depending if they are geometrically including (Hansen *et al.*, 2001) or closest to the gaze point (Monden *et al.*, 2005) or among a few nearest objects (Xu *et al.*, 2008). Smoothing and down-sampling the results over multiple frames partially compensates for runaway or missing mappings due to the aforementioned noise sources. The noisy raw gaze and object recognition data can thus be transformed into a largely coherent stream of fixation probabilities (Mehlmann *et al.*, 2014a).

TECHNOLOGICAL
& BIOLOGICAL
NOISE SOURCES

Even though the aforementioned noise sources can be suppressed to some degree, they can not be completely ruled out but must appropriately be handled by the agent's fusion engine. For example, when resolving a verbal reference with the user's gaze, the fusion engine can compensate for inaccuracies and irregularities by applying a quantification constraint over a time window which is temporarily aligned to the spoken referring expression. Such a quantification can, for example, be used to find the object that matches the user's verbal description and has been mapped to the majority of the user's gaze points around this time. This object is then selected as the most probable referent, even if other objects have also been looked at or have erroneously been reported as gaze targets during this period of time.

COMPENSATING
NOISE WITH
QUANTIFICATION

An example for such a situation can be found in scene ③ of the introductory scenario in Section 1.2 and is illustrated in Figure 3.3.2. In this scene, Marley refers to a specific photo on the workspace by saying "Tell me! Where is this beach?" and looking at the photo. Charly disambiguates this reference by considering those photos of which he knows that they show a beach and that Marley looked at within a certain time window. The time-line in Figure 3.3.2 depicts Marley's spoken utterance (Figure 3.3.2 ④), the computed gaze fixations (Figure 3.3.2 ⑤), and the real gaze positions on the table (Figure 3.3.2 ⑥). Between t_0 and t_7 Marley mainly looks at positions that are closest to photo 2, however, her gaze also shortly switches intentionally or subconsciously to the photos 1 and 3. Furthermore, between t_5 and t_6 the



recognition module is losing the tracking of photo 2 and therefore falsely reports photo 3 as the most probable gaze target during that time interval. Between time point t_0 and t_7 Marley is asking the question "Tell me! where is this beach?". In this, the deictic expression "this" occurs roughly at t_{this} and Charly takes the time window between t_1 and t_4 , an interval of about two seconds around t_2 , which is about 800 milliseconds before t_{this} (Meyer *et al.*, 1998; Griffin and Bock, 2000; Griffin, 2001; Kaur *et al.*, 2003), as basis for the disambiguation of the verbal reference. During this time interval, the photos 1 and 2 are both reported three times while photo 3 is only reported two times as gaze target. However, since only the photo 2 and 3 are showing beaches and photo 2 is more often looked at than photo 3, the conditional quantification returns photo 2 as the photo that Marley might most likely have referred to.

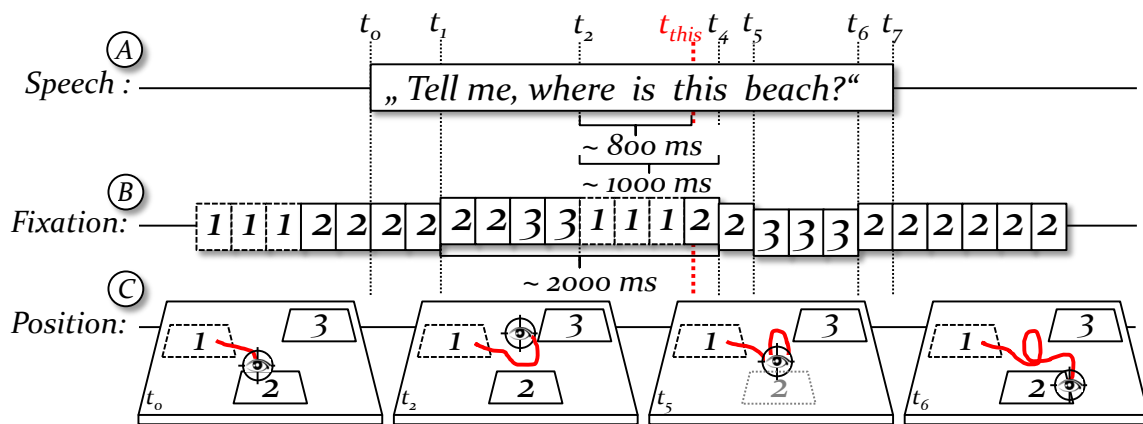


Figure 3.3.2: An illustration of a quantification of gaze events related to a verbal reference expression.

A social agent's multi-modal fusion engine needs to reason on application-specific semantic knowledge about the domain, task, user, and the physical environment. Moreover, the example illustrates that it must evaluate quantitative and qualitative temporal and ordering constraints between single input events as well as quantification constraints over sets of events. As later explained in Chapter 5, the proposed modeling approach meets these requirements with the *BFQL* that is implemented as declarative, embedded, domain-specific language (van Deursen *et al.*, 2000; Kosar and Mernik, 2006) in *PROLOG* (Kowalski, 1974; Emden and Kowalski, 1976; Kowalski, 1979; Clocksin and Mellish, 1981; Wielemaker *et al.*, 2012). The *BFQL* comprises a multi-modal event fusion calculus with logic predicates evaluating quantitative and qualitative temporal as well as ordering constraints between events. It comprises meta- or higher-order-predicates that implement generalized quantifiers (Colmerauer, 1978) and solution collection mechanisms (Naish, 1996). It includes a procedural part with dynamic predicates for the insertion and retraction of events and can be extended with application-specific facts and rules.

MULTI-MODAL
FUSION &
REASONING

3.4 Creating Behavior and Dialog Content

As mentioned in Section 3.1 and shown in Table 3.1.1, the third important modeling task is the creation of expressive and natural multi-modal behavior as well as credible and competent

dialog content. The first requirement is the *versatile composition of behavior* into behavioral activities consisting of individual, well-aligned behavioral units. This includes actions and nonverbal cues in a single modality, verbal statements, multi-modal utterances, and entire scenes, interleaving the multi-modal contributions of multiple agents. The second requirement is the *flexible integration of knowledge* into the agents' behaviors and dialog lines. This involves an easily manageable mechanism for the parameterization or substitution of particular parts of a behavioral activity. It allows the hybrid creation of behavior and dialog contributions by combining manually scripted with automatically generated contents, concluded from an agent's knowledge. Finally, the third requirement is the *automatic variability of behavior* using reasonable selection and alternation mechanisms for multi-modal utterances and scene activities. This avoids repetitive behavior and dialog content which would certainly make the agents' behavior appear less natural, believable, and competent.

3.4.1 Versatile Composition of Behavior

DIVERSE
BEHAVIORAL
CAPABILITIES

Humans exchange messages and signals via multiple modalities, such as speech, gaze, gestures, postures, and facial expressions. In physically situated, joint activities, they additionally communicate by manipulating and performing actions on objects in the environment. Rather than using single such actions or behaviors one by one, they usually use multiple of them in different modalities and communication channels at the same time (Jaimes and Sebe, 2007; Oviatt, 2012). They use their whole behavioral repertoire to produce well-aligned compositions of multi-modal behaviors, thus, improving interpersonal coordination (Bernieri and Rosenthal, 1991) and maintaining the conversational and perceptual ground (Clark and Brennan, 1991). For example, they use single gaze movements when following each other's attention (Argyle and Cook, 1976) or head movements and facial expressions when producing back-channel cues (Kendon, 1967; Yngve, 1970), but also composed multi-modal behavioral patterns and utterances, when eliciting feedback behaviors (Kendon, 1967; Bavelas et al., 2002), or producing multi-modal references through the coordinated use of speech, gaze and pointing gestures (Brennan, 1998; Hanna and Brennan, 2007; Kennington et al., 2015).

TECHNICAL
PROGRESS AND
DIVERSITY

In symmetry to the challenges for multi-modal fusion engines due to the growing field of sensor technology, the recent technical advancement in virtual environments and social robotics places new demands on the behavioral skills of social agents. The technical capabilities of advanced next-generation agent platforms must be very diverse and highly specialized, ranging from very simplistic designs that master only a small set of platform-specific commands to humanoid robots whose behavioral capabilities come, at least partly, close to the human skill repertoire (Fong et al., 2003; Vinayagamoorthy et al., 2006; Leite et al., 2013). Social agents capable of reproducing the human behavioral repertoire must therefore master a broad bandwidth of the above mentioned individual behaviors and versatile behavior compositions.

Examples of more or less complex compositions of behaviors can be found in various scenes of the introductory scenario in Section 1.2. Simple gaze movements can be found in scene ① in which Charly is following Marley's gaze shifts to the photos on the surface table that have caught her attention. A multi-modal behavioral pattern can be found in scene ⑤, in which



Charly is looking attentively to Marley by simultaneously raising his eyebrows and putting his head aside. Another one can be observed in scene ⑥, in which Charly performs a short smiling expression to mimic Marley's emotional display and then immediately averts his eyes and head to balance their interpersonal intimacy. A similar pattern is used in scene ⑩ when Charly closely aligns his head nod with a movement of his eyebrows to create a back-channel cue signaling interest and encouraging Marley to continue. Finally, a multi-modal utterance can be found in scene ④, in which Charly refers to a photo closely aligning a verbal reference in form of a deictic expression in the question "Where was that?" with his directed gaze to a photo in order to direct Marley's attention to this specific photo.

The above examples and other situations of the introductory scenario in Section 1.2, clearly demonstrate that a social agent has to master many diverse, properly aligned, multi-modal compositions of behavior to come close to the human behavioral repertoire. This includes individual platform-specific commands, nonverbal cues, verbal statements, multi-modal patterns and utterances. As explained in Chapter 5, the proposed modeling approach meets these requirements with a specially designed specification method, called *behavioral activities*, which can be specified with minimal effort and expert knowledge using the *Behavior Flow Script Language (BFSL)*. They resemble parts of a screenplay or movie script consisting of multi-modal stage directions for the alignment of verbal statements with co-verbal behaviors, actions, and commands. Special *scene activities*, pooled in *scene scripts*, are used to specify the interleaving of multiple agents' multi-modal behaviors and dialog contributions.

VERSATILE
COMPOSITION
OF BEHAVIOR

3.4.2 Flexible Integration of Knowledge

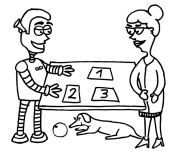
The aforementioned messages and signals that are exchanged between the participants of social joint activities can carry information that has been inferred from, or refers to their interaction context and domain knowledge which has or shall be incorporated into the common conversational ground (Clark and Brennan, 1991; Clark, 2005). In the course of a social interaction, the interaction partners remember each other's past dialog contributions and knowledge about their partners as well as the objects and events in the environment and uptake individual information from the available knowledge sources, such as the dialog domain, the common task, their interaction partners, and the discourse history (Flycht-Eriksson, 1999; Flycht-Eriksson and Jönsson, 2000; Wahlster, 2006) later during the interaction. This is the fundamental prerequisite for the human participants to have a coherent conversation, maintain the common ground and, thus, to make an informed, competent, and credible impression on the interaction partners (Clark and Marshall, 1981; Clark and Brennan, 1991).

CREDIBLE &
COMPETENT
BEHAVIOR

The maintenance of the common ground during collaborative tasks, like in the introductory scenario in Section 1.2, depends crucially on the successful exchange of knowledge about the objects that take center stage in the interaction (Clark and Marshall, 1981; Clark and Brennan, 1991; Brennan, 1998). For example, in a joint object sorting or manipulation task on a shared workspace, human speakers try their best to optimally differentiate the next object to be sorted, or worked on together, from possible alternatives, using a preferably unambiguous description of its characteristics. Vice versa, listeners fall back on the same mutual knowledge

REFERENCES
& MUTUAL
KNOWLEDGE

to resolve such references (Ros *et al.*, 2010; Mutlu *et al.*, 2013; Kennington *et al.*, 2015). They slip these descriptions in their utterances when referring to this object in order to minimize the collaborative effort to agree upon the next steps of the common task.



An example for the integration of context and domain knowledge into the dialog content can be found in scene ③ of the introductory scenario in Section 1.2. In this scene, Marley asks for information about the scenery which has been captured on a particular photo on the table by asking “Where is this beach?” and looking on this specific photo. In response, Charly uses his knowledge about the photo to report the time, place, and circumstances under which it has been taken by answering “This was your trip through France in 1980?”. His utterance is not entirely scripted by hand, but, is produced in a hybrid way, by enriching manually scripted kind of gap-text with photo-specific semantic information, for example, the year in which it has been taken. On the other hand, as shown in scene ④, Charly also tries to learn more about photos of which he has not yet enough information with the aim to ground the mutual knowledge about these photos. Thus, he is able to link the gathered information with the individual photos and to reproduce it the next time he is asked for it.

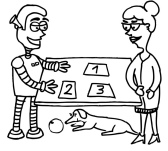
FLEXIBLE INTEGRATION OF KNOWLEDGE The above example makes evident, that a social agent must have the ability to embed inferred context and domain knowledge into its behaviors and dialog contributions, either through its partial integration into manually scripted content, or the delegation of the entire content creation to an external module. As explained in Chapter 5, the proposed modeling approach meets these requirements because the template-based scene format allows the substitution of *placeholder variables* with arguments, holding automatically generated or inferred content, using appropriate *BFGL* playback commands in a *BFSC*. Furthermore, it allows the *in-line insertion* of *BFSC* variables into behavioral activity specifications. This allows the hybrid creation of multi-modal behavior and dialog content, combining manually scripted, rapidly prototyped content with knowledge that has automatically and more sophisticatedly be generated by extern reasoning, recommender, or planning systems. This approach helps to create informed dialog content and make the agent appear more competent and credible while minimizing the needed initialization data, expert knowledge, configuration, and authoring effort.

3.4.3 Automatic Variability of Behavior

ANSWERING CLARIFICATION REQUESTS Miscommunication and the consequential disruptions of the common ground in conversations and joint activities mainly arise from misunderstandings, misconceptions, and non-understanding (Hirst *et al.*, 1994). These, in turn, are usually caused by ambiguous utterances, missing attention, or whenever one of the participants presumes sensory, perceptive, or cognitive abilities that the other cannot serve with (Gabsdil, 2003). A common error-handling strategy to face these types of miscommunication are clarification subdialogs in which listeners use confirmation requests to make sure that they correctly understood their partner or repetition requests to signal their misunderstanding or inattention (Allwood *et al.*, 1993; Traum, 1994; Clark, 1996; Purver *et al.*, 2003). In response, speakers usually repeat their utterance to repair the conversational ground when they notice that they have not been

understood correctly. If they recognize that a simple reproduction of content is not sufficient to improve their partners' understanding, then they go a further step by rephrasing or reformulating their statements. Such an intentional linguistic variation in a clarification dialog can efficiently contribute to the reestablishment of the common ground with minimal collaborative effort after a disruption (Clark, 1996, 2005; Clark and Wilkes-Gibbs, 1986).

An example of such a short clarification subdialog consisting of a repetition request and the reformulation of a misunderstood question can be found in scene ⑨ of the introductory scenario in Section 1.2. In this scene, Charly notices that Marley did not fully understand his question if she would like to have a cup of tea because she asks for a clarification with a repetition request "Get what?". The reason for Marley's misunderstanding is that she is deflected by a photo on the workspace and therefore interrupts Charly by saying "I miss these old times!" in scene ⑧ after Charly has already interrupted her before by asking "Marley, should I get a cup ..." in scene ⑦. As reaction, Charly repeats his question, however, he uses a variation by slightly rephrasing it instead of simply literally reproducing what he said before. This next time, he asks "Would you like some tea?" and Marley understands this second formulation and declines the offer by answering "No, thanks!". After she has rejected Charly's proposal, both agree on the same conversational topic again and the common ground has been repaired.



In addition to minimizing the communicative effort and therefore facilitating the grounding process (Clark and Brennan, 1991), people naturally use linguistic variations, that means variants of socio-linguistically significant morphological, lexical, and syntactical features (Biber, 1988; Chambers *et al.*, 2004), in order to express a variety of social meanings, such as intentions, beliefs, emotions, politeness, attitudes, and personality (Mairesse and Walker, 2011). Consequently, the creation of expressive, credible, and competent behavior, in general, requires the possibility for the natural variation of behaviors and dialog contents. The most simple thinkable method for linguistic variation is to allow, at least, the randomized avoidance of literal repetitions in semantically identical sentences (van Deemter *et al.*, 2005). Monotonous replications of the same utterances would otherwise make the agent appear less believable or clumsy and make the interaction boring (Cassell *et al.*, 2000b).

AVOIDING
REPETITIVE
BEHAVIOR

An example for the natural linguistic variation of dialog content in order to appear credible and competent can be found in scene ② of the introductory scenario in Section 1.2. In this scene, Charly uses different formulations of basically the same question, such as "Shall I tell you about that?" or "Are you interested in this?", whenever he asks Marley if she wishes more information about those photos that catch her attention for a longer period of time. He even combines the automatic linguistic variation with a semantical variation by integrating his knowledge about the content of the photos when proactively providing information about them, for example, by saying "This was in France!". Thus, he prevents repetitive behavior or making the same statement again and again, and avoids to appear clumsy or simple-minded.



The above examples demonstrate that a reasonable, purposive, and natural variation of behavior and dialog content in natural social interactions contributes to the grounding process and makes the interaction partner appear vivid, credible, and competent. Equipping a social

AUTOMATIC
VARIABILITY
OF BEHAVIOR

agent with this ability requires the variability of behavior, that means the autonomous, or at least automatic, rephrasing or reformulation of specific dialog contents or whole multi-modal utterances. As later explained in Chapter 5, the proposed modeling approach tackles this problem with the possibility to aggregate multiple alternatives for the same scene activity to a *scene group* and select individual alternatives at runtime based on a randomized or linearized *blacklisting strategy*. The different alternatives allow expressing the same contribution or dialog act semantics in different ways using slight variations of the wording, synonymous terms, or different gestures with the same meaning, thus efficiently minimizing the authoring effort for realizing variations.

3.5 Summary and Conclusion

In this chapter, I identified and discussed the modeling tasks that an author is faced with when modeling the interactive behavior of artificially and socially intelligent agents, especially when focusing on interpersonal coordination and grounding behaviors. These tasks are, first, the close coordination of an agent's behavioral functions and processes, second, the proper integration of user input and context events, and, third, the creation of credible and natural behavior and dialog content. Based on several exemplary scenes from the introductory scenario in Chapter 1, I illustrate that social interactions and joint activities share some common characteristics that directly lead to the identification of a number of task-specific modeling requirements. The modeling language for each of these tasks must satisfy these requirements in order to allow an author to successfully master this task.

The first task includes the incremental and reciprocal meshing, the parallel and hierarchical structuring, and the interruption and coherent resumption of functions and processes. The second one requires the maintenance of knowledge and multi-modal events in a well-organized working memory using a uniform knowledge representation format as well as a multi-modal fusion and reasoning calculus for their integration based on semantic, temporal and quantification constraints. The third task comprises the versatile composition of behavior including the flexible integration of knowledge and automatic variability of behavior and dialog content. In Chapter 4, these task-specific modeling requirements will serve as comparison criteria in the review of related work on multi-modal fusion, behavior and interaction modeling as well as behavior description. The concepts and languages of the modeling approach proposed in this thesis, that are only briefly addressed in this chapter, will be explained in more detail in Chapter 5 and illustrated in Chapter 6 based on a realistic use case.

CHAPTER 4

RELATED WORK — BEHAVIORAL ASPECTS AND MODELING APPROACHES

In Chapter 2, I introduced the interactional phenomena, referred to as interpersonal coordination and grounding. I also presented the individual behavioral functions that contribute to them, such as joint attention, language understanding, turn-taking, back-channel eliciting, intimacy regulation, to name but a few. In this, I have concentrated on explaining the roles of different gaze behaviors and voice activity overlaps for these functions. In Chapter 3, I discussed several characteristics of human behavior in social interactions and identified a set of modeling tasks and requirements for a computational behavior and interaction modeling approach. These must be met in order to be able to integrate and coordinate all the behavioral functions that contribute to interpersonal coordination and grounding. Finally, I gave a brief overview of the modeling concepts that are combined in the novel modeling approach presented in this thesis to tackle these requirements.

The modeling concepts of the proposed modeling framework in this thesis are essentially inspired or influenced by related work from different research fields. Relevant related research includes, in the broadest sense, approaches for modeling, first, dialog, behavior, and interaction for conversational embodied agents and social robots, second, multi-modal fusion in human-agent interaction or multi-modal user interfaces, in general, and, last, multi-modal behavior specification and description. The design decisions mentioned in Chapter 3 have been made to compensate for particular shortcomings of individual of these approaches and thus take a step beyond the state-of-the-art behavior and interaction modeling frameworks.

In the remainder of this chapter, in Section 4.1, I discuss related work in human-agent-interaction in regard of its research goals and contents. Representatives of this group study interactional phenomena similar to interpersonal coordination and grounding or focus on individual behavioral functions or sub-concepts of those regarded in this thesis. Afterwards, in Section 4.2, I present a variety of related modeling approaches that rely on more or less similar or closely related formalisms as the approach proposed in this thesis. I discuss, from a technical perspective, in how far these approaches offer the expressive power and practicality to meet the tasks and requirements identified in Chapter 3.

4.1 Modeling Behavioral Functions

Despite substantial research effort during the last decades, related work has not yet managed to develop modeling languages and authoring tools that allow integrating and coordinating the behavioral aspects of interpersonal coordination and grounding in an agent's behavior and interaction model. Most research investigates the one or the other role of gaze behavior or voice activity for these conversational phenomena in isolation. They focus on individual sub-aspects of interpersonal coordination and grounding or specific behavioral functions only. Among those, some even study these behavioral aspects exclusively in face-to-face conversations and do not consider physically situated joint activities in which the environment and the therein existing objects, persons, and events are an essential part of the interaction. Even worse, other colleagues developed only completely non-interactive, one-sided and thus inadequate models of gaze behavior that do not react to the user's behavior. Finally, some evaluate or ground their models based on observations made in over-controlled and simplified, experimental setups that have absolutely nothing to do with real social interactions.

4.1.1 Modeling Functions of Gaze Behavior

Earlier research on gaze behavior in the interaction of a human with an artificially intelligent agent works with *virtual characters* (Nakano *et al.*, 2003; Peters *et al.*, 2010; Bailly *et al.*, 2010; Pfeiffer-Lessmann *et al.*, 2012; Ruhland *et al.*, 2015). A well-known problem in the research on gaze behavior of virtual, animated, embodied conversational characters, that are rendered on two-dimensional displays, is similar to the phenomenon which is typically referred to as *Mona Lisa effect* (Rogers *et al.*, 2003; Moubayed *et al.*, 2013). Because the agent and user do not share the same physical space, it is hardly possible to see for the user where the agent is exactly looking to. That means, the agent cannot unambiguously establish mutual gaze with one of the participants in a multi-party interaction. Furthermore, it is difficult to identify the object that the agent is looking at in a physically situated interaction. This is not the case in immersive virtual environments and human-robot interaction in which human and agent are physically co-present in the environment, like in the introductory scenario in Section 1.2. So, this issue is not found in more recent related research which focuses on physically situated and collaborative joint activities with *social robots* (Imai *et al.*, 2003; Breazeal *et al.*, 2004; Doniec *et al.*, 2006; Huang and Thomaz, 2010, 2011; Huang and Mutlu, 2012; Huang *et al.*, 2015; Mutlu *et al.*, 2009, 2012, 2013; Staudte and Crocker, 2009, 2011).

Relevant related work can be found in the research on understanding, modeling, and evaluating the functions of social gaze behaviors in human-agent interaction (Srinivasan and Murphy, 2011; Srinivasan *et al.*, 2014; Ruhland *et al.*, 2015). The methodical means and outcome of these efforts are heavily depending on the research goals in mind, which can be human-, technology- or design-focused (Admoni and Scassellati, 2017). Here, I mostly consider heuristic, that means, literature-driven, design-focused approaches that aim at developing gaze models for specific tasks of the interaction, such as conversation or collaboration.

Joint Attention and Grounding

Much research studies the roles of gaze behavior for *joint attention* or, in better words, *shared attention* in physically situated interaction and social joint activities. Joint attention is here understood as the partners' ability to follow and direct each other's gaze to objects, to events, and to themselves with the aim to share a common point of reference (Mundy and Newell, 2007), not true cognitive joint attention (Kaplan and Hafner, 2006). The importance of joint attention has been shown by various experiments, in particular for collaborative joint actions between humans and robots on shared workspaces (Mutlu *et al.*, 2013). There, it was shown that, if participants pay more attention, and are aware of being looked at, they can recall more information (Mutlu *et al.*, 2006; Huang and Mutlu, 2012) which has a positive effect on task performance and completion.

As one of the first, Nakano *et al.* (2003) proposed a simple, incremental model of nonverbal grounding for an embodied conversational agent that describes a route on a map to a user. The model explains how head nods as back-channels, and gaze behaviors for joint attention, can be judged as grounding evidence. The agent checks the user's nonverbal signals during an utterance and afterwards continues monitoring the user's nonverbal cues until it gets enough evidence of understanding or non-understanding. Finally, the agent tries to judge the quality of the common ground based on the received evidence. For example, a verbal confirmation, head nod, or expected attentive gaze at the map can be interpreted as an acknowledgment of understanding. In a study they especially showed, that if the user keeps looking at the agent instead of following its gaze is an evidence of non-understanding and an attempt to evoke additional explanation from the agent. In other words, attention following to referred objects on the map suggests the user's understanding and cooperation.

Pfeiffer-Lessmann *et al.* (2012) developed a model of joint attention for an immersive virtual reality environment. As attempt to cover cognitive joint attention, their model defines four phases, characterized by the partners' mental states. These are, first, the initiate phase, second, the respond phase, third, the feedback phase, and, fourth, the focus-state in which both focus the object of attention and are mutually aware of the joint attention. Using this model, they investigated the ideal timing of an initiator's referential gaze to best introduce the target of the joint attention during the initiation phase. They also investigate which response times of the addressee for a referential act are perceived as acceptable and successful. As a result, they found that humans highly accept the virtual partner when it is using a natural timing of gaze to direct their attention, as found in human interactions.

A similar model for joint attention was developed by Huang and Thomaz (2011). It covers three parts, first, initiating, second, responding to, and, third, ensuring joint attention. To initiate joint attention, an agent uses addressing strategies including eye gaze, pointing gestures, and utterances. Afterwards, it periodically ensures joint attention by checking whether or not joint attention is reached and selects another addressing strategy if not. Therefore, it is constantly monitoring the partner's focus by looking back and forth. Using this model, the authors found that the behavior of a robot that is responding to joint attention is perceived as

more transparent, competent, and socially interactive. Moreover, a robot that is continually ensuring joint attention appears more natural and improves the task performance (Huang and Thomaz, 2010, 2011).

Engagement and Connectivity

Another interactional phenomenon that can be regarded as sub-aspect of interpersonal coordination and grounding is *engagement* (Glas and Pelachaud, 2015). It can be imagined as a dynamic, reciprocal process by which the interaction partners establish, maintain, and end their perceived connection during a social interaction (Sidner *et al.*, 2005). Besides bidirectional social cues, such as back-channels and adjacency pairs, also shared and mutual gaze are very important for engagement (Rich *et al.*, 2010; Holroyd *et al.*, 2011). It has been shown that a robot that moves its head towards and away from the speaker during a conversation can signal engagement (Sidner *et al.*, 2005). Based on these findings, Holroyd *et al.* (2011) developed a behavior model that supports the engagement between a human and a social robot through the establishment of shared and mutual facial gaze (Rich *et al.*, 2010).

Very similar, Kuno *et al.* (2007b) observed that a robotic museum guide that moves its head towards the visitor during its explanation increases the visitors engagement. They developed a behavior model that additionally coordinates a robot's head turning and gaze at transition-relevant points (Sacks *et al.*, 1974). They showed that a robot that also switches its gaze between the listener and an object of interest at these transition-relevant points results in even greater nonverbal engagement of the visitors (Kuno *et al.*, 2007a; Yamazaki *et al.*, 2008). The user's nonverbal engagement manifested itself in head turns and gaze movements towards the robot that revealed very precise timings and durations (Jefferson, 1973).

Hall *et al.* (2014) studied the perception of a robot's engagement by a user that reads a series of instructions to the robot. This robot's behavior model uses eye blinking, nodding responses, and, in particular, recurring gaze aversion and reengaging gaze to express engagement. While the user reads an instruction, the robot frequently averts its gaze for about one second in a random direction but tries to establish mutual gaze and performs a head nod in a speech pause. A study showed that nodding has a strong positive effect on engagement whereas gaze aversion could be interpreted as engaged thinking but also as boredom or inattention. Thus, depending on timing and accompanying behaviors, it could have a disengaging, detrimental impact (Doherty-Sneddon and Phelps, 2005) and must be precisely timed and directed to purposefully contribute to the perception of engagement (Yamazaki *et al.*, 2008).

Multi-Modal Disambiguation

Humans can with a fairly high precision determine where a robot is looking at in an interaction at a shared workspace (Moubayed *et al.*, 2013). However, it is one thing to determine the target of the gaze as an isolated task, but another to use it for facilitating comprehension of ambiguous language. Several researchers studied this role of gaze behavior for multi-modal disambiguation and reference resolution (Prasov and Chai, 2008; Ros *et al.*, 2010; Staudte and

Crocker, 2011; Boucher *et al.*, 2012; Okumura *et al.*, 2013; Prasov and Chai, 2010). For example, in their research on referential grounding in human robot interaction, Ros *et al.* (2010) found that visual perspective taking, which requires attention and gaze following, is necessary for the generation of appropriate referring expressions and their correct disambiguation. A robot must be able to reason on the user's focus of attention and the placement of object with respect to the user's vision in order to be able to compute whether an object is in the user's focus of attention, field of view, or out of the user's field of view (Ros *et al.*, 2010).

Vice-versa, Staudte and Crocker (2011) conducted experiments in which subjects watched videos of a robot that was describing objects on a table while gazing at them. They showed that the subjects were able to utilize the robot's gaze direction for successfully resolving referring expressions. Thus, exploiting the agent's focus of visual attention via gaze-following has a positive influence on utterance comprehension to anticipate, ground, and disambiguate spoken references (Staudte and Crocker, 2009; Staudte, 2010). In very similar experiments, also Boucher *et al.* (2012) showed that users can identify objects faster when the agent is gazing at these objects while referring to them. The other way round, errors in an agent's gaze hinder speech understanding, because people expect the agent's gaze to indicate what it intends to verbally reference. In cognitively more demanding tasks, incongruence in an agent's eye gaze and speech leads to bad task performance (Staudte and Crocker, 2009, 2011).

Turn-Taking and Role-Footing

Other aspects of interpersonal coordination and grounding, that have been studied mainly in isolation, are the regulation of the dialog structure and the footing of participant roles in human-agent interaction. Many models for these aspects incorporate findings from social psychology (Kendon, 1967; Duncan, 1972, 1974) to simulate the role of gaze in turn-taking. These approaches usually generated special gaze behaviors with the beginnings and ends of turns and utterances (Cassell *et al.*, 1999; Lee *et al.*, 2007) to produce turn-taking and floor management actions (Traum and Rickel, 2002; Bohus and Horvitz, 2010c). In these models, the agent looks away from the hearers at the beginning of a long turn and looks toward an addressee at the end of the turn to yield or assign the turn. During very short turns, it often looks toward an addressee throughout the whole turn, whereas, in between utterances of the same turn, it looks away from the partners to hold the turn (Lee *et al.*, 2007).

Mutlu *et al.* (2012) investigated the roles of gaze for conversation management, especially, how a robot can establish the participant roles, namely addressee, bystander, and overhearer, of its conversational partners using certain gaze cues (Mutlu *et al.*, 2009). They developed a footing model (Goffman, 1979) in which the agent signals the partners their assigned participant roles in different dialog phases, such as greeting, conversation, and turn-exchange. During greetings, the agent first welcomed all partners and at the transition to the conversation it diverted the gaze towards the intended addressee and away from the bystander. During the conversation it used different gaze distributions in which it looked most of the time to the addressee, seldom to the environment and only sporadically to the bystander, mostly in very short, acknowledging glances. It produced turn-yielding signals only for addressees

when exchanging the turn by using directed gaze at the end of utterances.

[Andrist et al. \(2014\)](#) developed a behavior model that is able to use gaze aversion to effectively regulate turn-taking. The model uses a timing statistics based on which it determines the frequency, length, start, and end times of an agent's gaze aversion, relative to its utterances. It is informed about the current conversational state, the start time, and length of upcoming planned utterances, and continuously plans and generates appropriate gaze behaviors in real-time. They found, that by averting gaze at the appropriate time, an agent more effectively held the conversational floor than when using gaze aversion at inappropriate times or not at all. Their results show that virtual agents that need to pause in their speech, for example, to process information or plan the next utterance, can use gaze aversion to hold the conversational floor and indicate that the next utterance is forthcoming.

Social Attitude and Personality

[Bee et al. \(2009\)](#) studied the roles of gaze behavior for the initiation of contact and regulation of intimacy between a human and an agent. They investigated which gaze signals an agent should convey in order to increase the user's interest and willingness to engage in an interaction with the agent. Their work is inspired by [Givens \(1978\)](#) who divide the progress of interpersonal encounters into discrete phases, such as attention, recognition, interaction, and others. They implemented a behavior model which allows varying the probability that the agent initiates gaze interaction using parameters, such as the maximal and minimal mutual gaze duration, the maximal gaze aversion duration, and the time the user must respond mutual for the agent to engage in a conversation. The agent using this model was perceived as more natural and engaged compared to a non-interactive agent.

[Bee et al. \(2010b\)](#) also studied the role of an agent's gaze behaviors for the perception of its personality. Their work is based on a gaze model by [Fukayama et al. \(2002\)](#) which allows controlling the frequency and duration of gaze movements to the user and gaze aversion to points in the environment. It can be parameterized by means of the amount of gaze to the user, the mean duration of a gaze, and gaze points during averted gaze. [Fukayama et al. \(2002\)](#) already found that a medium amount of gaze and a mean duration between 500 to 1000 milliseconds conveys a friendly gaze behavior. [Bee et al. \(2010b\)](#) evaluated different parameterizations of the model to study gaze-based dominance of an embodied conversational character. They showed that the parameterization can influence the perception of the agent's personality traits, such as dominance, extroversion, and agreeableness.

Since the model used by [Bee et al. \(2010b\)](#) does not respond to the user's gaze, they extended it to an interactive model ([Bee et al., 2010c](#)). They introduced further parameters for the maximal and minimal duration of mutual gaze in reaction to the user's current gaze and the time how long the agent waits until the user must respond with mutual gaze. The interactive model is more realistic since user and agent recognize and coordinate each other's gaze. It was used to investigate the agent's interactive gaze behavior's influence on the user's experience in terms of, for example, rapport, engagement, attraction, and social presence. Even

though being interactive, the model is still restricted because it does not include the interplay of gaze with other modalities. It furthermore ignores its role in turn-taking, attention following, back-channel eliciting, and other behavioral functions. Nevertheless, it can serve as orientation how to model gaze distributions that can influence aspects like intimacy or dominance in different dialog phases or participant roles.

4.1.2 Modeling Functions of Voice Overlaps

An example of research that focuses on the roles of voice overlaps and interruption attempts in interactions with social agents is that of [Cafaro et al. \(2016\)](#). This work investigated the effects of different interruption types and strategies in a conversation between a speaker agent and an interrupting agent on a human observer. They varied the amount of overlap between speakers and compared a disruptive with a cooperative interruption strategy. They found that agents that choose to attempt a cooperative interrupt are perceived as more engaged. The amount of overlap influenced the perception of both agents' social attitude and personality, such as friendliness and dominance. These are useful results that can be utilized in a turn-taking or interruption handling model implemented using our approach. However, the agents in these experiments were not interactive but instead rendered as opposing silhouettes in a picture giving a visual hint that two entities are talking to each other. The models and applications developed with the approach proposed in this thesis overcome this shortcoming by moving beyond videos and involving the user as an interlocutor in the interaction.

A social agent whose behavior and interaction model allows a context-sensitive handling of a user's interruptions has been presented by [Crook et al. \(2010\)](#). It enables the agent to quickly detect and intelligently respond to different kinds of user interruptions. In this, it takes account of the meaning of the user's utterance and the user's emotional state. The model distinguishes interruptions from other acoustic phenomena such as extraneous noises or verbal back-channels based on the duration of the voice overlap and the intensity of the user's voice. If the user is speaking long or high enough, then the agent immediately stops speaking and shows a surprising nonverbal behavior, such as facial expression and a back-channel, for a timely reaction. In parallel, it processes the user's utterance and considers semantic and prosodic information to perform a reasonable and empathic linguistic response. The recovery from an interruption can then be achieved by resuming to the argument that the agent was making immediately before the interruption, or, by addressing the content of the interruption. The information from the affective analysis flows into the agent's reaction, for example, by mirroring the user's emotional state with a facial expression and tone of voice. This sophisticated model was a good inspiration for the design of interruption handling strategies using our approach.

[Chao and Thomaz \(2011\)](#) developed a model for the control and analysis of timing aspects in a social agent's turn-taking behavior. They conducted a study of the effects of interruptions on turn-taking dynamics in a scenario in which the user and a robot are cooperating to solve the Towers of Hanoi problem. The robot interrupts its actions and looks immediately to the user whenever the user's hand starts moving to the shared workspace. In addition, the

robot immediately interrupts its speech if the human performs an action before the robot has finished an action request. Later they extended their turn-taking model such that it allows parameterizing the participants' interruptibility as well as various timing parameters (Chao, 2012; Chao *et al.*, 2014). They used the model to research to which extent the dynamic change of different parameter settings results in different social dynamics (Thomaz and Chao, 2011; Chao and Thomaz, 2013; Smith *et al.*, 2015). They showed that interruptions lead to increased task efficiency due to increased user initiative. This improved the overall feeling of interaction balance and produced a higher sense of fluency. They also recognized that not only speech overlaps but also actions and resource conflicts on shared workspaces can lead to turn-taking conflicts. Both findings influenced the development of the modeling approach and the turn-taking policy in the illustrative model in Chapter 6 of this thesis.

Another sophisticated, fully-fledged turn-taking model of a social agent was developed by (Thórisson, 2002; Thórisson *et al.*, 2010). The model joins two major approaches to turn management, first, the signal-based approach (Duncan, 1972), and, second, the opportunities-based approach (Sacks *et al.*, 1974). Similar to the interruption handling model of Crook *et al.* (2010), it uses a layered architecture with several parallel update loops operating at different speeds. A reactive layer is responsible for ideomotor actions, like looking away when taking or holding the turn (Goodwin, 1981) while a control layer coordinates mental activities, such as utterance starts, stops, and interrupts. The layers consist of concurrent modules that are responsible for perception, multi-modal integration and decision-making. The turn-taking decisions are encoded using rules that implement theories from behavioral sciences. The model includes dynamically adjustable parameters for impatience, willingness to give turn and eagerness to speak. The layered architecture design of this model influenced the development of the modeling approach and the illustrative model in Chapter 6 of this thesis.

4.2 Modeling Tasks and Requirements

From a technical perspective, relevant related work aims at the development of generally usable modeling languages for interaction and behavior management, multi-modal fusion, and behavior specification. Thus, this work primarily pursues a universal approach to individual modeling tasks without focusing on particular interactional phenomena, such as interpersonal coordination and grounding, or particular behavioral functions, such as attention, engagement, or turn-taking. Many of these modeling frameworks require a substantial degree of expert knowledge and programming skills, thus being unsuitable for non-experts.

4.2.1 Coordinating Functions and Processes

COMPARISON A social agent's behavior and interaction model is responsible for managing the interplay
CRITERION of the behavioral functions that contribute to interpersonal coordination and grounding.
CATALOG This includes the proper prioritization, synchronization, and interleaving of concurrent and nested computational and behavioral processes. As described in Chapters 1 and 3, the applied modeling approach must hence meet a number of criteria. First, it must manage the

incremental and reciprocal meshing of input processing, knowledge reasoning, and behavior generation. Second, it has to allow the *parallel and hierarchical structuring* of the model into processes and layers, and their coordination with synchronization and inter-process communication mechanisms. Third, it must enable the immediate *interruption and coherent resumption* of behavioral functions and processes to consistently respond to changing behavioral goals and interruptions. Related modeling frameworks that deal, in one way or another, with these requirements are used for interaction and dialog management in multi-modal user interfaces (Trung, 2006; Dumas et al., 2009b, 2010), embodied conversational agents, and social robots (Cassell et al., 2000b; Jung et al., 2011) as well as interactive digital storytelling (Cavazza et al., 2001, 2008). Aside from naive scripting approaches, they can be divided into the rather distantly related frame-, rule-, and plan-based approaches and the more closely related, versatile kinds of state-based and hybrid modeling approaches.

Rule-Based Approaches

The early *scripting* approaches, such as, for example, in the animated pedagogical agents COSMO or HERMANTHEBUG (Lester et al., 1997; Lester and Stone, 1997), simply compose sequences of text blocks, audio clips, and animations for specifying the behavior and dialog content of a social agent. However, a behavior and interaction model must prescribe the agent's behavioral reactions to possible situations and user inputs and provide sufficient variations in order to avoid predictive and repetitive behaviors. In *rule-based* approaches the agent's behavior and interaction model is encoded as a set of *rule operators*. An *inference engine* cyclically examines the *precondition* of each rule and selects a subset of those whose conditions are satisfied with regard to the current *working memory*. The *action* of one of those rules is then executed which may modify the working memory again and thus trigger the execution of other rules in the next cycles. The preconditions are used to specify user input constraints or context information states and actions execute system commands to generate agent behaviors and utterances. These approaches use rule-based programming systems, such as CLIPS (Riley, 1997) or JESS (Friedman-Hill, 2003), based on the RETE algorithm (Forgy, 1990), or logic inference systems, like PROLOG (Clocksin and Mellish, 1981) and its derivatives, like SWI-PROLOG (Wielemaker et al., 2012) or JINNI (Tarau, 1998).

FEATURES OF
RULE-BASED
APPROACHES

An example for a rule-based extension of a scripting approach for behavior and interaction modeling is IMPROV (Perlin and Goldberg, 1996). IMPROV was designed to allow creative experts who are not primarily programmers, such as artists and screenwriters, to create interactive performances with virtual actors. Therefore, an author specifies the agents' observable behavior with manually written scripts and writes rules governing their higher-level behavioral choices. These rules basically compute the probabilities for individual author-provided choices based on a weighted list of author-specified influence factors. They are used to determine when and with whom actors engage in conversations and their reactions to possible situations and user inputs in different contexts.

EXAMPLES OF
RULE-BASED
APPROACHES

Similar, SCREAM (Prendinger and Ishizuka, 2002; Prendinger et al., 2002, 2004) uses a user-extensible set of rules and facts for the emotion-based, high-level scripting of a character's

mind. The author uses designated features and dynamically updated facts to define an agent's profile in terms of initial goals, beliefs, and attitudes. A rule set makes up the reasoning component which represents the agent's mental processes and states that determine its behavioral responses to received communicative acts. The system can easily be extended by adding or modifying rules that encode the agent's cognitive processes, for example with affective appraisal rules.

Many other systems pursue a rule-based approach to model different aspects of an agent's behavior and interaction model. The authoring framework *CREACTOR* (Iurgel *et al.*, 2009; da Silva *et al.*, 2009) uses a rule-based method to define an agent's interactive performance. The conversational agent *REA* (Cassell *et al.*, 2000a) uses a rule-based high-level behavior control module which performs inferencing tasks that determine its deliberative communicative actions. The *ROBOCUP* commentator systems *BYRNE* and *MIKE* employ a rule-based approach for generating an agent's emotional state and reason about events (André *et al.*, 2000). The sport commentary agent *ERIC* (Strauss and Kipp, 2008) uses a rule-based approach for reasoning, affective appraisal and, template-based natural language generation.

DISCUSSION OF
RULE-BASED
APPROACHES

The examples demonstrate that rule-based systems have versatile application possibilities. With their working memory and powerful inference mechanism, rule-based systems are well suited in scenarios where knowledge reasoning is involved. This strength is, for example, utilized by tools like *MUDRA* (Hoste *et al.*, 2011), *MIDAS* (Scholliers *et al.*, 2011) or *HEPHAISTK* (Dumas *et al.*, 2009b,a, 2010, 2014) which use rules to build higher-level facts from low-level information for multi-modal integration and complex event processing. Rules can be chosen such that a fine-grained interleaving of processing, reasoning, and generation steps is, in principle, possible. It is, in theory, also possible to control parallel processes and their incremental and reciprocal meshing. However, with growing rule bases the procedural aspects of a rule-based system become very hard to maintain because the rule system becomes unwieldy and cumbersome and the possible states of the behavior and interaction space become hardly comprehensible. Even worse, conflicts that occur when more than one rule may be executed must often heuristically be resolved by particular meta-rules that express preferences regarding the priority of candidates on the rule agenda (Hayes-Roth, 1985).

Plan-Based Approaches

FEATURES OF
PLAN-BASED
APPROACHES

In *plan-based* systems for dialog and interaction management, an *automated planner* receives the current world or *information state*, a set of *plan operators* with *preconditions* and *effects* that change this state, and a *goal state*. The planner then searches for an *action sequence* that would, if successfully executed, bring the system from the current to the goal state. A classical example is *STRIPS* (Fikes and Nilsson, 1971) which represents a world model as a collection of first-order logic formulas and employs a resolution theorem prover that uses a means-end analysis strategy in the goal-based search for the desired goal-satisfying action sequence.

REACTIVE
PLANNING

Planning the course of a social agent's dialog and interaction is helpful to ensure a coherent interaction and accomplish the goals of a joint activity. However, while humans maintain and

tend to follow plans once they have one, they can also change them on the fly when needed by dropping an intention or changing a part of the plan. This phenomenon is referred to as *practical reasoning* (Bratman, 1987) and is characteristic for interpersonal coordinating and grounding. Similar, it is impossible to plan an agent's whole interaction in advance when the partners can interrupt each other, respond freely at every turn, and are exposed to environmental events that constantly change their information states. Thus, many automated planners provide a more appropriate and practical reasoning strategy that is referred to as *reactive planning* (Georgeff and Ingrand, 1989).

Most reactive planners for dialog and interaction management make use of the *hierarchical task network* (HTN) style of plan operators (Erol *et al.*, 1994; Nothdurft *et al.*, 2015). The goal of the HTN planner is to produce a sequence of actions for a *task*, which is either *primitive*, that means a plan operator, or *compound*, consisting of subtasks and a method that prescribes its decomposition into subtasks. The planner uses the method to decompose compound tasks into smaller and smaller pieces until it reaches primitive, executable tasks. Typical examples for HTN planners are *O-PLAN* (Currie and Tate, 1991) or *SHOP* (Nau *et al.*, 1999) and *SHOP2* (Nau *et al.*, 2003). Using HTN structures can ensure that the plan follows a stipulated courses of action and reduce the search space (Freedman, 2000). They can, for example, resemble the hierarchical structure of task-oriented dialogs (Grosz and Sidner, 1986) such that the context given by the hierarchy information can be used for resolving referring expressions. HTN plans can, on the one hand, be continuously planned and generated at run-time and, on the other hand, explicitly be provided as pre-authored hierarchical and-or trees.

HIERARCHICAL
TASK NETWORK

An early hierarchical planning component is used in *IMP* (André and Rist, 2000, 2001) for generating the behaviors of presentation teams of several agents. It decomposes a presentation goal into elementary goals, thus producing a dialog script consisting of the *dialog acts* (Bunt *et al.*, 2010; Bunt, 2011) to be executed by the single agents and their temporal order. One of the first reactive, hierarchical planner is *APE*, an integrated planning and execution system at the heart of the *ATLAS* dialog management system (Freedman, 2000). It controls a mixed-initiative dialog between a human user and an agent and can handle arbitrarily nested discourse constructs, making it more powerful than dialog managers based on flat finite-state machines.

DIALOG
PLANNING
EXAMPLES

Other hierarchical approaches were taken in the applications of the *SMARTKOM* project (Herzog *et al.*, 2004; Herzog and Reithinger, 2006) and in the dialog manager of the *VIRTUALHUMAN* system (Löckelt *et al.*, 2007). Here, each individual agent is autonomously controlled by a conversational dialog engine whose action planner works with a hierarchical interaction structure. In this hierarchy, *dialog acts* are the atomic communication units, their rule-governed exchanges create short *dialog games* (Carlson, 1985; Levin and Moore., 1977), such as question-answer pairs, which are then combined to activities to implement an agent's goal-directed behavior.

The dialog management system *D4G* (Rich and Sidner, 2012) combines HTNs with *dialog*

trees (Despain, 2008). It uses a *HTN* to model the high-level hierarchical task or goal structure of a dialog. Relatively small subdialog trees are then attached at the fringe of the *HTN*. *D4G* builds on *DISCO* and *COLLAGEN* (Rich and Sidner, 1998; Rich et al., 2001; Rich and Sidner, 2012) and their collaborative discourse theory (Grosz and Sidner, 1986). Another extension of *DISCO* is the *DTASK* system (Bickmore et al., 2009, 2011) which uses *HTN* with *adjacency pairs* (Schegloff, 1968; Schegloff and Sacks, 1973), which are pairs of a single agent utterance with a user response menu, at the fringe of the network, instead of complete dialog trees, as in *D4G*. Yet another variant applies rules at the live nodes of a *HTN* to generate dialog candidates (Rich et al., 2002).

Other plan-based dialog managers that use *HTNs* are *RAVENCLAW* (Bohus and Rudnicky, 2003, 2009), its predecessor *AGENDA* (Xu and Rudnicky, 2000), and its extension *OLYMPUS* (Raux and Eskenazi, 2007). They model domain-specific aspects of the dialog control logic via a *HTN* which is manually constructed by an author. A network's inner nodes control the execution of their subordinate nodes and thus represent the temporal and logical dialog structure. The leaf nodes represent atomic dialog actions or dialog moves, such as output production, information requests, or system actions. Such a dialog model's execution is then performed by a domain-independent dialog engine and influenced by the *HTN*'s dialog logic and the inputs of the users.

NARRATION PLANNING EXAMPLES A variety of reactive *HTN* planning approaches have been applied for the dynamic creation of narrations for interactive digital storytelling. For example, Cavazza et al. (2002) combine a *HTN* planner with reactive agent behaviors to cope with unexpected events and situations. Each individual character's behavior in the story is initially defined by an author as *HTN*. Interactions of the user with the agent or the environment can produce situations that lead to action failures and cause the re-planning of the agents' goals in the network. Similar approaches are used in the *MIMESIS* (Riedl et al., 2003) and *FACADE* (Mateas and Stern, 2003a) systems to account for user interactions that might require a re-organization of the narrative an interactive drama. In *IDTENSION* (Szilas, 2003) and *FEARNOT!* (Aylett et al., 2006), plans are generated at run-time rather than being explicitly represented as pre-built goal trees.

DISCUSSION OF PLAN-BASED APPROACHES The examples show, the major strength of *HTN* is their hierarchical structure which may be generated or pre-authored. Making the hierarchical task structure of the dialog explicit facilitates the development and maintenance of large dialogs. It allows automatically generating great parts of the dialog model or reusing authored structures and improves the control over the dialog's evolution. However, the actual contribution exchange at the fringe of the network is still modeled highly idiosyncratic. Reactivity, by means of re-planning of deliberative dialog contributions is possible with reactive planning approaches. Reactive, hierarchical planners, eventually enriched with statistical techniques (Ultes et al., 2017; Budzianowski et al., 2017), are certainly a good choice for planning large and nested dialog structures that involve reasoning tasks. However, they are not well suited for a fine-grained interleaving of processes and modeling ideomotor or highly reactive behaviors. Similar to rule-based approaches, it is laborious to control and synchronize parallel processes with them.

State-Based Approaches

Most similar to the modeling approach proposed in this thesis are various kinds of *state-based* approaches to model the procedural aspects of an agent's behavior, interaction, and dialog. They rely on different finite-state-based formalisms in which an agent's mind, dialog, and interaction logic, or individual behavioral aspects are represented by means of states and transitions. Some of them allow hierarchically refining and decomposing the model into parallel components. Others provide very unique and specialized modeling constructs, such as particular nodes, edges, commands, guards, policies, histories, and others. Part of this state-based family of modeling approaches are different types of augmented or extended state transition networks, state-chart dialects, and Petri-net variants.

FEATURES OF
STATE-BASED
APPROACHES

An early authoring system with a finite-state-based approach to dialog and interaction management is the *CSLU* toolkit (Sutton and Cole, 1997; McTear, 1998, 1999). It comes with a graphical modeling environment which facilitates the development of spoken dialog systems by non-experts. This allows visually assembling finite-state-based dialog models by linking together different graphical dialog objects. These objects can be used for generating prompts, recording and recognizing speech input, or performing system actions. The dialog models may include branching decisions, loops, jumps, and subdialogs that permit a hierarchical, modular model design and the reuse of already modeled parts.

HIERARCHICAL
TRANSITION
NETWORKS

Similar tools are *DIALOGOS* (Bobbert and Wolska, 2007) and *DIAMANT* (Fliedner and Bobbert, 2003) which have been used for the development of spoken or text-based dialogs with robots (Koller and Kruiff, 2004). They use extended state transition diagrams consisting of different nodes represented as icons on a graphical workspace. Those include nodes for input processing, output production, script execution, and sub-graph. These are called and parameterized like functions and have local memories and return values. However, they have no history mechanism and restricted real-time capabilities because they cannot be interrupted.

Another extended state transition network is used by the *SCENEMAKER* authoring suite (Gebhard *et al.*, 2003a) to control the dialog behavior of interactive agents in the *CROSSTALK* (Rist *et al.*, 2002; Baldes *et al.*, 2002) and *COHIBIT* (Ndiaye *et al.*, 2005) systems. Nodes may have pre-scripted scenes attached that specify the agents' dialog lines and co-verbal behaviors. Special super nodes may contain nested networks to create a hierarchical structure. They provide conditional, probabilistic, scheduled and, in particular, interrupting transitions. However, they do not use an interaction history to consistently resume interrupted super nodes.

Many other approaches use hierarchical state transition networks to model social agents' interaction and dialog behavior (Person *et al.*, 2000; Graesser *et al.*, 2004, 2005; D'Mello and Graesser, 2013) or narratives in interactive digital storytelling (Speerling *et al.*, 2006). However, they do not allow the parallel decomposition of the model. This challenge is, for example, tackled with the successor version of *SCENEMAKER* (Mehlmann, 2009; Gebhard *et al.*, 2012), called *VSM*, using new authoring concepts, such as parallel networks, variable scoping, interruption policies, and a history mechanism. Sticking with the visual modeling paradigm states can only be activated using transitions, such that authors do not need programming

PARALLEL
TRANSITION
NETWORKS

experience but must eventually cope with less clearly arranged graphs.

Another type of parallel transition network focusing on the representation of parallelism and synchronization are the *PAT-NETS* (Cassell *et al.*, 1994; Badler *et al.*, 1995) that have been used in the virtual presenter character *JACK* (Noma and Badler, 1997; Noma *et al.*, 2000). *PAT-NETS* are concurrently executed, parallel finite-state automata which are created using coordination rules. They can call actions in the simulation, make conditional and probabilistic transitions or can sleep until a desired time or a specific condition is met. They can synchronize with each other by waiting on shared semaphores and can even invoke or kill each other. They are used for the combined control of high-level behavior and low-level animations, for example, to model the interaction between several agents and the synchronization of gaze and hand movements to the dialog for each individual agent.

EXAMPLES OF
STATE-CHART
DIALECTS

Other approaches to dialog and interaction management adopt or extend classical state-chart variants (Harel, 1987; Harel *et al.*, 1990; Harel and Naamad, 1996; Harel and Politi, 1998; von der Beeck, 1994; Drusinsky, 2004; Harel and Kugler, 2004; Drusinsky, 2006; Crane and Dingel, 2007). An example is *TQS* (Traum *et al.*, 2008; Gandhe and Traum, 2008; Gandhe *et al.*, 2008) that partly relies on state-charts for rapidly authoring dialog capabilities of virtual humans (Gandhe *et al.*, 2009). Similar to an earlier approach (Kronlid and Lager, 2007), its dialog manager follows the *information-state model* (Traum and Larsson, 2003) using update rules that are implemented as state-charts using *SCXML*, a *W3C*¹ working draft for describing state-charts (Barnett *et al.*, 2007). Attempting to render *Harel State-Charts* (Harel, 1987; Harel and Politi, 1998) *SCXML* provides modeling concepts for hierarchy, concurrency, history and broadcast communication.

SCXML is also used in *DEAL* (Brusk *et al.*, 2007; Hjalmarsson *et al.*, 2007; Wik *et al.*, 2007) to describe a game's interaction structure and model the behavior and dialog of non-player characters (*NPCs*). *DEAL* models each *NPC* as a separate state-chart, running in parallel with each other and the game world. Parallel state-charts are also used to separately model different behavioral aspects of the *NPCs*, such as their states-of-mind, verbal behaviors, as well as accompanying conversational gestures and emotional expressions.

State-chart variants have also been used to model turn-taking, joint attention, and other interactional phenomena, apart from dialog flow management. They have, for example, been used to model the original *SSJ* model (Sacks *et al.*, 1974) for turn-taking in dyadic interactions (Kronlid, 2006, 2008). The *IRISTK* toolkit (Skantze and Moubayed, 2012; Skantze and Johansson, 2015) uses its own state-chart variant, based on *Harel State-Charts* (Harel, 1987; Harel and Politi, 1998) for the rapid authoring of an interactive robot's dialog behavior in different applications (Moubayed *et al.*, 2012, 2013). It is, for example, used to manipulate several parameters in turn-taking management, such as syntactic completeness and filled pauses in a robot's speech, as well as facial gestures, breathing patterns, and gaze behaviors to deal with processing delays in the system (Skantze *et al.*, 2014, 2015).

¹<http://www.w3.org/TR/scxml/>

Other state-based approaches for dialog and interaction management are special *Petri-net* variants (Murata, 1989; Genrich, 1991). Many are designed to incorporate facilities for timing, hierarchy, parallelism, and synchronization. For example, the *DISCO* and *COLLAGEN* dialog manager series (Rich and Sidner, 1998; Rich *et al.*, 2001; Rich and Sidner, 2012) has been extended to the *DISCO-RT* framework (Nooraei *et al.*, 2014) using a Petri-net-based approach for the synchronization of parallel, multi-modal behaviors and arbitration between conflicting behaviors (Holroyd *et al.*, 2011).

EXAMPLES OF
PETRI NET
VARIANTS

The *CADENCE* modeling framework (Chao *et al.*, 2014; Chao and Thomaz, 2016) uses a Petri-net variant, called *TPNs* (Chao and Thomaz, 2011; Chao, 2012), for controlling the autonomous, multi-modal turn-taking behavior of a social robot in a dyadic, collaborative interaction (Chao, 2015; Chao and Thomaz, 2016). *TPNs* are designed to explicitly represent temporal constraints, concurrent processes, synchronization, and interruptible behavior execution. They were used to develop a parameterized turn-taking model (Chao and Thomaz, 2013; Smith *et al.*, 2015) to research how the dynamic change of different parameter settings results in different social dynamics (Thomaz and Chao, 2011). *CADENCE* has similar objectives and solution concepts as the approach proposed in this thesis, and should be usable to tackle very similar problems. However, it has, so far, been used to model turn-taking and interruption handling, only, but not for other behavioral functions contributing to interpersonal coordination and grounding.

Hybrid and Agent-Systems

Hybrid modeling approaches combine the aforementioned techniques to manage each behavioral aspects with the most appropriate technology. This helps to balance the differing needs for modeling the aspects of reactive and deliberative behaviors and their interplay with dialog and interaction management. For example, procedural and ideomotor aspects of behavior are often modeled with state-based approaches while plan-based techniques are used for knowledge reasoning tasks involved in dialog management. A combination of rule-based and state-based systems is often used to reduce the visual complexity of state transition diagrams with complementary rules. Furthermore, such hybrid approaches are also used to unite input processing and interaction management in a uniform formalism.

FEATURES OF
HYBRID
APPROACHES

Such a hybrid modeling approach can be found in the virtual training environment *MRE* (Swartout *et al.*, 2001, 2006). In *MRE*, each agent's behavior is modeled using a control method that best fits its character type. Agents with limited behavioral range are scripted in beforehand and their behavior scripts are triggered based on environmental events. Characters that interact with the trainee are controlled with reactive plan-based modules (Marsella *et al.*, 2000; Johnson *et al.*, 2000; Rickel and Johnson, 2000) to master a broader bandwidth of socially competent behavior and deal with unanticipated situations. The plot of a training session is modeled with a finite-state-based formalism that allows author-defined deviations from the main storyline while keeping control of the overall flow.

EXAMPLES OF
HYBRID
APPROACHES

Another example is the authoring tool *CYRANUS* (Iurgel, 2004, 2006) which has been used in

ART-E-FACT (Iurgel, 2004), *VIRTUAL HUMAN* (Goebel *et al.*, 2007), and *INSCAPE* (Balet, 2007; Dade-Robertson, 2007). It relies on hierarchical transition networks comprising reference states defined outside the network. Each reference state integrates a rule-based engine to activate it without a transition when the network would otherwise become too dense or the transition conditions too complex. This allows a possibly more clearly arranged transition network but is a renunciation from the visual modeling paradigm and requires a certain amount of programming expertise. A similar approach is pursued in *ITEACH* which runs a finite-state machine and a rule-based system in parallel and synchronizes them using shared variables (Miksatko and Kipp, 2009; Miksatko *et al.*, 2010).

Instead of combining state machines with forward-chaining rule systems, the *MUDIS* (Giuliani *et al.*, 2008) interaction and dialog manager integrates finite-state machines (Mealy, 1995) and a *PROLOG* (Clocksin and Mellish, 1981) inference engine. *PROLOG* is used for knowledge reasoning in the dialog steps that are modeled as reusable, application-independent dialog states with the underlying finite-state machine. Similar to the approach proposed in this thesis, the states call commands to the agent's text-to-speech and behavior generation components and transitions are guarded with logic queries.

An agent-based approach is *ASAP* (Kopp *et al.*, 2014), a middle ware for *ECAs*, that is tailored to the incremental interleaving of input processing and behavior generation for a fluid and smooth interaction flow. Its distributed architecture comprises concurrent components that asynchronously and incrementally communicate using multi-directional message protocols based on *BML*, *FML* and *PML* (Kopp *et al.*, 2006; Vilhjálmsson *et al.*, 2007; Scherer *et al.*, 2012). The applications developed with *ASAP* mainly focus on the verbal and gestural aspects of fluid conversation management, such as dialog planning, adapting to verbal and gestural feedback, turn-taking and interruption handling. They do not consider gaze behaviors or other behavioral functions that are important in physically situated joint activities.

4.2.2 Integrating Input and Context Events

COMPARISON
CRITERION
CATALOG

Interpersonal coordination and grounding presupposes the robust understanding of the partners' behaviors. This includes the integration of information from input events distributed over multiple modalities and context knowledge in a social agent's multi-modal fusion engine. As discussed in Chapters 1 and 3, the fusion and reasoning formalism must meet several criteria to achieve this goal. First, it must represent the possibly irregular and heterogeneous events with a *uniform representation format* to avoid compatibility and idiosyncrasy issues. Second, application knowledge and input must be maintained in a *well-organized working memory* that preserves the event's actual chronological order across turns. Third, it must provide an expressive and uniform formalism for *multi-modal fusion and reasoning* that allows integrating application knowledge and partial input information based on temporal, semantic, and quantification constraints. Finally, with regard to the liaison with the behavior and interaction management, it must support the fine-grained, *step-wise fusion of inputs* for handling continuous interactions and incremental processing. Relevant related effort on multi-modal fusion engines that tackles these requirements is found in interaction modeling

frameworks and interactive social agents (Benoit *et al.*, 2000; Oviatt *et al.*, 2000; Lalanne *et al.*, 2009; Turk, 2014; Caschera *et al.*, 2015; Oviatt *et al.*, 2017).

Early Fusion Approaches

The very first approaches to multi-modal fusion are attempts to combine natural language input and pointing actions for spatial tasks in map-based, intelligent user interfaces. In these very early systems, multi-modal integration is primarily realized through the procedural extension of an existing speech or text understanding engine. They don't allow the simultaneous input of gestures and text but required the user to consecutively alternate between the keyboard and the pointing device. Gesturally enriched, natural language input is based on a sequential, one-to-one mapping of deictic expressions or markers in the spoken or typed text to textual descriptions of entities pointed to on the screen. The referent is simply replaced with the object description provided by the gesture recognition without semantically analyzing the pointing gesture and its consistency with the natural language input.

NAIVE EARLY
APPROACHES

An example of such a system is *SCHOLAR* (Carbonell, 1970), a geography tutoring system which enables the use of pointing gestures to select regions on maps displayed on a screen. The *NLG* system (Brown *et al.*, 1979) allowed sequentially mixing natural language descriptions with pointing on a touch screen to draw simple geometric objects. The *SDMS* system (Bolt, 1980) allowed creating and manipulating geometric objects by natural language and coordinated pointing gestures. In the *NLMENU* system (Thompson, 1986), the user used a mouse to rubber band areas on a map while giving verbal commands. Similar, the *SHOPTALK* system (Cohen *et al.*, 1989; Cohen, 1991) allowed inserting syntactic representations of referents, selected using pointing gestures, into natural-language queries and commands.

Among the early approaches are also slightly more sophisticated systems that overcome some restrictions of the very first, naive systems. First, they cope with the simultaneous use of natural language and gestural input within the scope of a single turn. Second, they support a greater range of gestural input, that means graphic gestures in addition to just deictic pointing gestures. Finally, they check the semantic consistency of speech input with accompanying gestural input. An example for such a system is *XTRA* (Kobsa *et al.*, 1986; Allgayer *et al.*, 1989), a multi-modal interface to expert systems, which combines natural language, graphics, and pointing for input. It enables the user to multi-modally refer to objects on the screen even with underspecified descriptions or imprecise pointing gestures. Similar, the *CUBRICON* system (Neal *et al.*, 1989) can resolve inconsistencies between pointing gestures and accompanying verbal expressions by applying semantic constraints (Neal *et al.*, 1988).

ADVANCED
EARLY
APPROACHES

To sum up, none of the early prototypes uses a uniform representation format or maintains a turn-overarching working memory. Consequently, most can not integrate continuous, truly parallel, verbal and nonverbal input using temporal or quantification constraints. In contrast, the user's hands have to move back-and-forth from the keyboard to the pointing device in order to switch between typed and gestural input. Typed words and gestures must occur in the exact order before the multi-modally created sentence can be parsed as a whole. Multi-

DISCUSSION
OF EARLY
APPROACHES

modal fusion is restricted to a single turn and it is not possible to incrementally interleave input processing and output generation. Speech is always treated as an indispensable modality while gestures are treated as a secondary dependent mode. Multi-modal integration can only be triggered by the appearance of a deictic reference expression in the speech stream. Its resolution with the information from a pointing gestures is only in a few systems guarded by a semantic consistency checks between the gestural and spoken input interpretation.

Frame-Based Approaches

FEATURES OF FRAME-BASED APPROACHES

Multi-modal semantic fusion requires methods and algorithms that allow combining meaning representations from individual modalities, such as speech, gesture, gaze, and others, into an overall interpretation. In the above discussed early fusion engines, each modality had its own such meaning representation framework and an idiosyncratic method was used for multi-modal integration. However, a more flexible and extensible semantic fusion mechanism requires a uniform meaning representation format for all modalities and a well-defined, universal combination operation for partial meanings. In *frame-based* fusion engines, unimodal parsers translate input from individual modalities in isolation to a common intermediate representation format, called *semantic frames* (Minski, 1975; Fikes and Kehler, 1985). Multi-modal integration is then achieved by merging these modality-specific frames to obtain a combined interpretation. Frames are a uniform semantic representation framework and basis of various domain-specific formats, such as *EMMA* (Johnston, 2009), *M3L* (Herzog et al., 2004), *MMIL* (Kumar and Romary, 2003) and others (Nigay and Coutaz, 1993, 1995; Wasinger et al., 2005; Wasinger, 2006). The actual appearance of semantic frames, the merging algorithms, and fusion architectures, however, still remain rather idiosyncratic and application-specific, such that they are hardly comparable to each other.

EXAMPLES OF FRAME-BASED APPROACHES

Koons et al. (1993) developed one of the first frame-based systems in which the user may use various types of gestures, directed gaze, and speech commands to modify the contents of a two-dimensional map. In this system, the user's multi-modal inputs are parsed to a common frame-based structure which are assigned timestamps and modality-specific, semantic contents. The timing information is used to realign them to their real chronological sequence before merging them. The merging algorithm is based on very application-specific evaluation methods that, among other tasks, are responsible for reference resolution.

The *MATIS* system is a multi-modal interface to an air traffic information database which allows combining natural language input, graphical input, and direct manipulations with a mouse. It uses a very specific fusion architecture with two-dimensionally structured frames representing the semantic and temporal information of a user's input (Nigay and Coutaz, 1995). They can be fused based on criteria such as their semantic complementarity, temporal relations, and context information. All modalities are equivalent according to the *CARE* properties (Coutaz et al., 1995) such that the user may use any modality for triggering a command (Bouchet et al., 2004).

The *MSA* system is a mobile shopping assistant in an instrumented environment that inte-

grates speech- and gesture-based interactions on a mobile device and with real world shopping products (Wasinger *et al.*, 2005; Wasinger, 2006). Its semantic frame representation organizes user's input by means of the discourse segment type, input modality, action type, a confidence value, begin and end timestamps, and other pieces of information. The modality fusion process is influenced by re-weighting and scoring of confidence values, consideration of time frames, and semantic consistency.

The only approach that uses a domain-independent merging algorithm is presented in the *JEANIE* system, a multi-modal appointment scheduling calendar that integrates speech and pen input (Vo and Wood, 1996; Vo and Waibel, 1997). Modality-specific input is parsed to partially filled frames which are then merged to a combined interpretation by creating the union of each slot's value set and adding the respective confidence values. Nested frames are merged recursively thus creating aggregate frames, encoding alternative interpretation hypotheses. This uniform merging technique can handle high-level information from arbitrary modalities in a generic manner and is thus modular and extensible.

Summarizing, frame-based approaches are, similar to rule-based ones, a very heterogeneous group of fusion frameworks. What binds them is that each of them has a uniform meaning representation format which, however, can significantly differ from that of others. Even worse, their fusion architectures and merging algorithms are often idiosyncratic. Nevertheless, they all can realign the user's multi-modal behaviors, occurring simultaneously in a single turn, to their chronological sequence using a short-term memory. Then consider modality-specific confidence values, timestamps, and semantic constituency constraints for temporal and semantic consistency checks. In turn, they do not work closely intermeshed with the behavior and interaction management, such that incremental fusion and interleaving with reasoning and generation processes is basically not possible.

*DISCUSSION OF
FRAME-BASED
APPROACHES*

Unification-Based Fusion

The inconsistencies concerning the fusion architectures and algorithms of frame-based approaches led to the attempt to unitize both using *unification-based* fusion engines. These frameworks achieve multi-modal integration through the *unification of feature structures* (Kasper and Rounds, 1986, 1990) or *typed feature structures* (Kay, 1979; Carpenter, 1992) that serve as uniform meaning representation. Feature structure unification combines two feature structures to a single new one which then contains all the information of the original two but nothing more. It is a well-defined operational framework that has turned out to be especially suited to the task of multi-modal fusion because it does not only allow combining complementary or redundant input, but also ruling out or overlaying contradictory input based on structural and content-related constraints (Johnston *et al.*, 1997; Alexandersson and Becker, 2003; Ehlen and Johnston, 2013).

*FEATURES OF
UNIFICATION-
BASED FUSION*

Unification-based fusion has, for example, been applied for the integration of spoken commands with gestural input in the pen- and voice-controlled *QUICKSET* systems (Cohen *et al.*, 1997a,b; Johnston *et al.*, 1997; Wu *et al.*, 1999). These are interactive military training simu-

*EXAMPLES OF
UNIFICATION-
BASED FUSION*

lations on mobile devices and desktop computers in which the users may use a pen to draw lines, areas, or symbols on the map and simultaneously speak commands to lay down certain units or objects. In these systems, multi-modality caused a significant speed increase (Cohen *et al.*, 1998, 2000) and enabled the mutual compensation of recognition errors in the individual modalities (Oviatt, 1999). Later, the unification-based fusion engine of *QUICKSET* was extended with a statistical approach for selecting among multiple possible combinations of speech and gesture based on determining the optimal weights for combining their posterior probabilities (Wu *et al.*, 1999). This hybrid, symbolic-statistical architecture achieved an even more robust functioning, compared with the original approach alone (Wu *et al.*, 1999, 2002; Kaiser and Cohen, 2002; Kaiser *et al.*, 2003) by complementing the temporal and semantic constraints with a statistical evaluation.

PROGRESS TO UNIFICATION-BASED PARSING Unification-based approaches are, on the one hand, based on a well-understood formalism and suited for elementary tasks, but, on the other hand, rather inflexible since they solely rely on the unification operation of single feature structures (Johnston, 1998b). This operation is, however, only able to combine two partial meanings, such as a single spoken element and a single gesture, to a multi-modal combination. In order to account for a broader range of multi-modal expressions, and to be applicable for more complex problems, such as the incremental recognition and fusion of variable event constellations, more general *unification-based parsing* approaches have been pursued (Johnston, 1998a,b; Johnston and Bangalore, 2000; Giuliani and Knoll, 2007; Bangalore and Johnston, 2009). Therefore, the systems based on the basic unification operation have been augmented with constraint-based reasoning to operate declaratively in a multi-modal integrator often using a *unification-based multi-modal chart-parser* (Earley, 1970) that has been fed with a *multi-modal grammar* (Johnston, 1998a,b). There exists a wide variety of constraint- and unification-based multi-modal parsing approaches with rather specific parsing methodologies (Johnston, 1998b; Holzapfel *et al.*, 2004; Stiefelhagen *et al.*, 2004; Sun *et al.*, 2007, 2009; Lukas *et al.*, 2010).

DISCUSSION OF UNIFICATION-BASED FUSION Multi-modal fusion engines that rely on the unification and overlay of feature structures or unification- and constraint-based parsing methods have a successful history in multi-modal interfaces and interactive systems with social agents. Their generic nature enables them to integrate the parallel use of arbitrary multi-modal inputs based on semantic and temporal constraints. Thus, they can handle versatile multi-modal request and command styles, however, they do not scale well. On the one hand, the simple variant can only combine two inputs. On the other hand, the parsing approaches usually suffer from great computational complexity. Finally, the specification of unification-based multi-modal grammars is a cumbersome and laborious process. Consequently, while multi-modal parsers are theoretically able to return partial parsing results, the incremental interleaving of input processing and output generation remains hardly achievable with these approaches.

Rule-Based Approaches

FEATURES OF RULE-BASED APPROACHES Several of the aforementioned frame- and unification-based parsing approaches can be considered as special cases of *rule-based* systems. They use the same meaning representation

frameworks and production rules of a multi-modal grammar or combination rules within a frame-merging algorithm. However, most of them solely rely on a single kind of constraint-based unification or merging operation for the production of combined or interim results. Other rule-based approaches try to achieve more flexibility by permitting an author to define his own, more specific, but again considerably idiosyncratic, integration rules relying on a rule-based programming system, such as *CLIPS* (Riley, 1997) or *JESS* (Friedman-Hill, 2003), or logic inference engines, like *PROLOG* and its derivatives (Clocksin and Mellish, 1981; Wielmaker *et al.*, 2012). Thus, they offer more possibilities of semantic and temporal reasoning as well as the integration of application knowledge into the fusion process.

An early fusion approach which employs a common, symbolic data representation scheme and production rule-based integration mechanism, based on the *CLIPS* system (Riley, 1997), was introduced by Sowa *et al.* (1999). A drawback of their approach is the growth of the computational complexity with the number of symbols that are available during rule matching. To cope with this problem, they use a time window approach in order to keep the symbol memory small and efficient. The relevance of a symbol decreases with time such that the fusion engine may retract it after some retention period. The time span of preserving symbols depends on their semantic content, such that more complex objects, that tend to be sparse compared with low-level symbols, will last over a longer time span. This concept of temporal persistence of an input or behavior can be found in the garbage collection mechanism of the approach proposed in this thesis.

EXAMPLES OF
RULE-BASED
APPROACHES

Holzapfel *et al.* (2004) present a rule-based multi-modal fusion approach similar to the above mentioned multi-chart parsers (Johnston, 1998b, 2000). Input events are asynchronously parsed into a semantic representation based on typed feature structures and added to an input set. A constraint-based parsing is performed on the input set in order to merge the different input streams. The parsing algorithm uses a special kind of multi-modal fusion rules to determine which inputs can be merged, added or removed, and instructions to construct the merge result. The parser supports a set of predefined constraint types, such as content-wise, time, and modality constraints. Additional constraints can be defined by a rule-writing author. The approach has, for example, been used to resolve ambiguities of the user's speech using gestures and head poses in human-robot interaction (Stiefelhagen *et al.*, 2004).

The *MUDRA* system (Hoste *et al.*, 2011) uses a rule-based approach to combine low-level data processing and high-level semantic inference. Multi-modal input is represented as semantic frames and stored in a common fact base. A rule-based system is used to extract and combine features from these frames at different abstraction levels. The language supports various constraints concerning the attributes, probabilities, and temporal properties of inputs from multiple users. A similar approach is pursued by the *HEPHAISTK* framework (Dumas *et al.*, 2009b, 2014, 2009a) which can be considered as unified frame-based approach to multi-modal fusion and interaction management. It uses *SMUIML* (Dumas *et al.*, 2010) to define XML-encoded rules that determine the temporal and semantic conditions for multi-modal fusion as well as the creation of output actions and dialog-context transitions. Both tools' capability to integrate input from different abstraction levels and processing stages has been

adopted by the approach proposed in this thesis.

Another rule-based multi-modal fusion approach is pursued in the *PATE* system which is used in the *COMIC* application (Pfleger, 2004). It uses a production rule system to be able to perform a context-based multi-modal integration by incorporating the current dialog state into the multi-modal fusion process. This, for example, allows comparing recognition results to the expectations anticipated by the current dialog state instead of solely relying on confidence values. Incoming data is stored as typed feature structures in a working memory and is assigned some weight which fades out with time. The fusion relies on unification and overlay operations on feature structures (Alexandersson and Becker, 2003) that determine the consistency of two data items and combine them if they are consistent. A uniform formalism and working memory for multi-modal input and application knowledge is likewise found in the approach proposed in this thesis. It allows the context-based, incremental fusion and fine-grained interleaving with the interaction and dialog management.

State-Based Approaches

*SIMPLE
STATE-BASED
APPROACHES*

The early state-based approaches have been developed because, on the one hand, they allow a light-weight implementation with low computational needs, on the other hand, they are easier understood and allow a rapid prototyping of multi-modal integration models. They use finite-state machines that are traversed as the user produces multi-modal input. States represent partially parsed results and transitions are taken when a certain symbolic input from a specific modality is received. Some approaches produce the semantics of the parsed input step-wise, while taking transitions, others produce them as a whole when reaching a final state. The declaration of a state-based model can be performed in a purely graphical way by drawing the labeled state transition diagram or declaratively in a multi-modal grammar from which it is then generated. Furthermore, they have the advantage that they can be generated from textual descriptions and be learned from observations.

A well-known finite-state-based approach for multi-modal recognition was presented by Johnston and Bangalore (2000) and used in the *Match* applications (Johnston *et al.*, 2002; Johnston and Bangalore, 2004) which are mobile, multi-modal city information systems that allow the combined use of handwritten pen input and freehand gestures. It relies on the compilation of a multi-modal, context-free grammar into a single, multi-tape, finite-state automaton (Mohri *et al.*, 2002) which simultaneously parses input lattices from multiple modalities and combines their content into a semantic representation, on the fly (Johnston and Bangalore, 2001, 2005). At that time, the automaton was compiled into a cascade of finite-state transducers which can compose directly with lattices from speech and gesture recognition and are usable with available finite-state processing tools (Bangalore and Johnston, 2000, 2009).

Also Bourguet (2002) proposed finite-state machines as a simple method for modeling multi-modal fusion and provides the appropriate rapid prototyping solution with the graphical editor *IMBUILDER* and the interpreter *MENGINE* (Bourguet, 2003a,b, 2004). She was among the first to discuss the possibility to automatically learn finite-state fusion models from observa-

tions of users (Bourguet, 2006). These should freely produce input sequences with the aim to activate specific functions of the application which are then used to learn finite-state machines accepting these inputs. Furthermore, she studies the option of generating finite-state fusion models from textual use case descriptions of possible interaction sequences (Chang and Bourguet, 2008; Bourguet and Chang, 2008).

In contrast to the laborious and computationally complex unification-based, multi-modal parsing approaches, the simple state-based approaches are easier to understand and have lower computational needs. While earlier approaches separate unimodal parsing and multi-modal integration into consecutive processing stages, they can step-wise integrate and understand parallel multi-modal inputs within a single user turn as one stage. However, instead of a uniform representation format, they use rather idiosyncratic symbolic data and workarounds to handle multi-dimensional data or a growing number of possible user inputs. Without working memory, they do not take account of modality-specific processing delays. Instead, they handle input events in the order they arrive in a specific modality during a single turn. They use simplistic timeout mechanisms to determine which gestures and speech inputs should be considered part of the same turn. They do not enable us to specify temporal or quantification constraints or to integrate application knowledge into the fusion process. Even though being state-based, they are not able to incrementally interleave behavior understanding, knowledge reasoning, and response generation.

DISCUSSION OF
STATE-BASED
METHODS

More advanced state-based modeling approaches were developed to meet the particular needs of interactive applications in virtual environments and physically situated joint activities with social agents. They consist of states and transitions that be accompanied with the generation of actions or events and can be augmented with guarding temporal and semantic constraints to control if a transition may be taken or not. States include local memory and a hierarchical structure can be created using nested state-machines in states or transitions. Thus, in addition to model the integration patterns of multi-modal input, they particularly allow modeling the interaction flow and the application context. They can be used for the parallel processing of incoming percepts on varying level of granularity, the partial and incremental parse execution and the latching into real-time applications. They support the multi-modal integration based on the inputs' semantic content, information from the application context level as well as parameterizable temporal and semantic relations. They allow the combination of discrete interactions which will be executed in one atomic operation as well as continuous interactions, such as gesture streams produced by kine-mimic gestures that takes over and control the interaction.

ADVANCED
STATE-BASED
APPROACHES

Such advanced state-based modeling approaches use, for example, special variants of *Petri nets* (Murata, 1989; Genrich, 1991) or specially designed *state transition network* (Wasserman, 1985; Hudson and Newell, 1992) for multi-modal fusion. For example, Navarre *et al.* (2005) and Ladry *et al.* (2009) describe a Petri-net-based approach, called *ICOs*, to model multi-modal fusion operations using high level Petri-nets (Murata, 1989; Genrich, 1991). A similar approach uses a *Temporal Augmented Transition Network (TATN)* (Latoschik, 2001, 2002, 2005). These advanced state-based approaches based with *TATNs* or *ICOs* cope with

issues such as, hierarchy, parallelism, synchronization, and incremental fusion of behaviors and action. They are consequently able to cope with discrete and continuous interaction and allow incremental parsings and interleaving of individual recognition and generation steps. They are suited to express temporal and semantic relations while being able to incrementally recognize behavioral patterns. However, so far, none of the presented advanced state-based fusion engines is able to evaluate quantification constraints as our proposed approach with the *BFQL*. Furthermore, they again use rather idiosyncratic, non-declarative expression languages to label states and transitions while the approach in this thesis relies on well-defined, declarative first- and higher order logic.

4.2.3 Creating Behavior and Dialog Content

COMPARISON
CRITERION
CATALOG

As described in Chapters 1 and 3, a description language for the specification of plausible and competent, multi-modal behavior and dialog content of a social agent must satisfy several criteria. First, it must allow specifying *versatile compositions of behavior*, ranging from individual system actions and unimodal cues over complex behavioral patterns to whole well-aligned multi-modal utterances. Second, it has to enable the *flexible integration of knowledge* into the agent's behaviors and dialog lines. Finally, it must support the *automatic variability of behavior* and linguistic variations. Related work has proposed different behavior description and specification languages as reusable representations of social agents' multi-modal behavior and dialog content on different abstraction levels. They range from high-level meaning representations to low-level, lexical descriptions of behaviors, their synchronization points, and expressivity parameters, such as spatial and temporal extent, repetitivity, power, or fluidity (Pelachaud, 2005). Some have also the role of communication protocols to separate and interface software modules that implement different functions in behavior creation, such as planning and realization engines (Kopp et al., 2006). Most relevant related work, however, focuses on multi-modal behavior descriptions as they are provided by a behavior generation and handed over to a behavior realization module (Kipp et al., 2010).

BEHAVIOR
GENERATION
WITH BEAT

An early approach to the generation of social agents' behavior is the *Behavior Expression Animation Toolkit (BEAT)* (Cassell et al., 2004). It aims at the separation of verbal behavior specification and the automated generation of co-verbal behavior. In *BEAT*, a human operator provides the typed text input that the agent is to speak. The resulting multi-modal utterance is represented in a *XML*-based format that can be processed by different animation engines. The co-verbal behaviors are aligned to the spoken utterance relying on rules derived from communication and behavior research. The system's output is a synthesized speech stream together with a number of synchronized nonverbal behaviors. The rule set may be extended with author-defined policies to adapt the approach to application-specific needs and achieve behavioral variations. The integration of application knowledge into the dialog content must have been done before the co-verbal behavior generation.

SEMANTIC
MODELING
WITH APML

The *Affective Presentation Markup Language (APML)* (Carolis et al., 2004; Pelachaud, 2005) is an example of a language that specifies an agent's behavior at the meaning level. It uses a taxonomy of communicative functions, defined as meaning-signal pairs, (Poggi et al., 2000),

such as information about the agent's belief, goal, affective or mental state. A meaning may be communicated with different signals and a signal may be used to convey different meanings, depending on the agent's personality, culture, or other factors. For example, an emphasis meaning may be expressed with a raise eyebrow, a head nod, or a combination of both signals. Vice versa, a raised eyebrow may be a sign of surprise, emphasis, or suggestion and a smile can express joy or be back-channel cue. The meaning-signal pairs of *APML* allow a lot of behavioral variation but *APML* is not laid out to integrate knowledge.

The *Multimodal Utterance Representation Markup Language (MURML)* is a constraint-based description language for an agent's nonverbal behaviors and their co-verbal alignment (Kopp *et al.*, 2003; Kopp and Wachsmuth, 2004). Each *MURML* specification contains a textual specification of the verbal part of an agent's the utterance including internal chunk borders markings. These are associated with additional specifications of para-verbal or nonverbal behaviors such as prosodic features, gestures, or facial animations. The cross-modal, temporal alignment of speech with co-verbal behaviors, is defined in terms of absolute times, relative to the start of a chunk, or in co-occurrence with linguistic elements. Gestures and their sub-movements, skeletal and morph target animations, and their synchronization points can narrowly be controlled by defining spatial and temporal constraints. In contrast to the generative, meaning-based approaches of *BEAT* and *APML*, the rather descriptive *MURML* provides authors with explicit, fine-grained control over the alignment and shaping of co-verbal behaviors. It furthermore allows the parameterization of specifications which can be used for knowledge integration and variation.

DESCRIPTIVE
APPROACH
WITH MURML

A hybrid approach is defined in the *SAIBA* framework where the behavior generation happens in several processing stages, which are intent planning, behavior planning and behavior realization (Kopp *et al.*, 2006). In this, the *Functional Markup Language (FML)* (Heylen *et al.*, 2008) is used to encode the communicative intent without referring to physical realization, similar to *BEAT* or *APML*. Then the *Behavior Markup Language (BML)* specifies the verbal utterance and nonverbal behaviors like gesture, posture and facial expression (Kopp *et al.*, 2006; Vilhjálmsón *et al.*, 2007), similar to *MURML*. By defining an additional application independent dictionary of behavior descriptions, called *gesticon* (Krenn and Pirker, 2004), the language distinguishes between abstract behavior definitions and concrete realizations.

HYBRID
MODELING
WITH SAIBA

Since languages like *BML* and *MURML* employ concepts like relative timing and lexical behavior descriptions, Heloir and Kipp (2010) argue that a number of low-level concepts should be further moved from these languages to what they call the declarative animation layer on an even lower level of abstraction. They justify the need for such an additional layer, as a thin wrapper around the platform-specific animation engine and below the higher-level behavior control layers, with abstracting away from implementation details while giving access to the functionality of the engine (Kipp *et al.*, 2010). An example for such an intermediate representation which can directly be executed by the animation engine are the scripts used in the *EMBR Embodied Agents Realizer (EMBR)* animation engine (Heloir and Kipp, 2009, 2010). They allow specifying gesturing, facial expressions, pose shifts, blushing, gaze control, and autonomous behaviors like breathing and blinking of virtual characters. They also over-

come the restriction of most *BML* realizers following a fixed-timing, pre-scheduling approach which is not ideally suited for event-driven, incremental behavior generation (Holroyd and Rich, 2012). This shortcoming has, however, been overcome with extensions to *BML*, such as *BMLA*, that allow the fluent, on-the-fly changes to ongoing behaviors in the respective realizers (van Welbergen *et al.*, 2012).

DISCUSSION &
CONCLUSION

The rather descriptive approaches of *MURML* and *BML* are closer to *BFSL* than the generative method of *BEAT* or the meaning-based approach of *APML*. Since *BML* is designed to be automatically created, it doesn't allow the template-based insertion of application knowledge as *BFSL* and *MURML*. Like *BFSL*, many template-based approaches have variability as one of their central design specifications. So, template-based approaches are not less variable, maintainable, and linguistically well-founded than generative approaches (van Deemter *et al.*, 2005) in general. Since templates in *MURML* and *BFSL* can be specified by hand, they might have advantages in some cases, for example, if no good linguistic rules are yet available in *BEAT* or sentences have no assignable meaning in *APML*. So, the manual description of behaviors by a schooled and experienced expert with an approach like *BFSL* can improve the quality of behavior and dialog content compared to a largely automated approach such as *BEAT* or *APML*. It better supports peoples, such as designers, artists, screenwriters, or psychologist, to exploit their professional expertise, experience and knowledge in their field of activity to create aesthetically attractive and highly goal-oriented behaviors and interactive presentations (Stone and Lester, 1996).

The behavior synchronization mechanisms of *BML* and *MURML* allow a more fine-grained and precise intra-personal behavior alignment than those of *BFSL*. For example, the different phases of gestures can be precisely aligned with the agents gaze behaviors and individual words in its spoken utterance using precisely timed synchronization points. In *BFSL* the start and stop of a gaze behavior or animation can, however, only be linked to the points in time at which the agent's speech synthesis has just finished producing a word. However, the co-verbal, intra-personal alignment method used by *BFSL* is certainly easier to understand and quicker to apply by non-computer experts. Furthermore, while all of the aforementioned approached use a more or less cumbersome *XML*-based syntax, *BFSL* allows specifying behavior with a much more intuitive style that comes very close to natural language descriptions. The experiences with this kind of specification format have been positive throughout and it has become evident that non-expert authors very easily understand and learn this behavior specification format. Thus, *BFSL* certainly finds a good balance between expressiveness and practicability.

If however necessary, the modeling framework in this thesis can easily be integrated with languages for the specification of behavior, emotion simulation and emotional speech synthesis such as *FML*, *BML* and *EMOTIONML* (Kopp *et al.*, 2006; Vilhjálmsón *et al.*, 2007; Heylen *et al.*, 2008; Kipp *et al.*, 2010; Schröder *et al.*, 2011). This can be achieved by implementing the respective user-defined playback commands and integrating them in the modeling framework proposed in this thesis. Furthermore, the same mechanism makes it possible to use specific commands that fall back on rules that can map abstract dialog acts from domain-

and application-specific dialog act taxonomies to utterances (Core and Allen, 1997; Bunt *et al.*, 2010; Bunt, 2011).

4.3 Summary and Conclusion

The literature survey in this chapter has shown that related work has not yet managed to develop modeling frameworks that allow integrating and coordinating the behavioral aspects of interpersonal coordination and grounding in a social agent's behavior and interaction model. The content-related comparison in Section 4.1 revealed that most related research focuses on individual behavioral functions, such as engagement (Rich *et al.*, 2010; Holroyd *et al.*, 2011; Hall *et al.*, 2014), joint attention (Huang and Thomaz, 2011; Pfeiffer-Lessmann *et al.*, 2012; Mutlu *et al.*, 2013), turn-taking (Lee *et al.*, 2007; Mutlu *et al.*, 2012; Andrist *et al.*, 2014), multi-modal disambiguation (Ros *et al.*, 2010; Staudte and Crocker, 2011; Boucher *et al.*, 2012), interruption handling (Chao and Thomaz, 2011; Crook *et al.*, 2012; Smith *et al.*, 2015), or intimacy regulation (Bee *et al.*, 2009, 2010b,c), in isolation, without regarding the interplay with the other aspects and the dialog management. However, these functions represent only individual sub-aspects of interpersonal coordination and grounding. In fact, they are highly interwoven such that they can not be treated in isolation but much rather call for a uniform modeling approach. It is neither sufficient to study behavioral functions in vitro, nor to simply develop isolated controllers for the individual behavioral aspects.

A more technical comparison in Section 4.2 showed that other work focuses on the development of modeling frameworks that are generally suited for mastering one specific modeling tasks. Among those are approaches to dialog and interaction modeling (Swartout *et al.*, 2006; Giuliani *et al.*, 2008; Gandhe *et al.*, 2009; Bohus and Rudnicky, 2009; Rich and Sidner, 2012; Kopp *et al.*, 2014; Chao and Thomaz, 2016), multi-modal fusion (Johnston *et al.*, 1997; Johnston, 1998b; Pflieger, 2004; Latoschik, 2005; Dumas *et al.*, 2009a; Hoste *et al.*, 2011), and behavior specification (Cassell *et al.*, 2004; Pelachaud, 2005; Kopp and Wachsmuth, 2004; Kopp *et al.*, 2006; Heloir and Kipp, 2010; van Welbergen *et al.*, 2012). However, they do not see the whole picture because the design and function of their modeling frameworks are not oriented towards and have not been examined for their suitability to model any of the interactional phenomena. Finally, the great majority of related work still investigates models for social interactive behavior under over-controlled or over-simplified laboratory conditions or do not even study interactive behavior at all, but evaluate only non-interactive models that do not react to the user's behaviors. These experiments and the behavioral models they utilize have often nothing to do with real social interactions and therefore have a very limited reliability and significance.

PART III

CONCEPTION AND ILLUSTRATION

*“Life is a lot like jazz —
it's best when you improvise.”*

GEORGE GERSHWIN

CHAPTER 5

CONCEPTION — DESIGNING THE BEHAVIOR FLOW MODELING LANGUAGE

In Chapters 2 and 3, I demonstrated that the key challenge for a behavior and interaction modeling approach is the control and coordination of the manifold behavioral functions and processes that contribute to interpersonal coordination and grounding. In Chapter 3, I identified the modeling tasks involved in facing this challenge, discussed the task-specific requirements, and briefly outlined my solution ideas from a language engineering perspective. The literature review in Chapter 4 showed that related work does not present sufficiently expressive and practicable modeling approaches to face the aforementioned challenge because they are either limited to specific behavioral aspects or focused on individual modeling tasks.

In this chapter, I present the theoretical foundations and conceptual design of a novel modeling approach, going beyond the aforementioned related efforts. It is the first modeling framework to combine the benefits of hierarchical and concurrent state-charts, logic programming and a template-based behavior specification format for modeling the interactive behavior of artificially intelligent social agents. With respect to *expressiveness*, it goes beyond the state-of-the-art because it masters the complex coordination and interplay of the many behavioral aspects that contribute to interpersonal coordination and grounding. In this, it has a remarkable *practicability* since it uses mostly declarative and visual modeling formalisms and uniform representation formats. Exploiting the modeling principles of modularity and compositionality, it allows the iterative and distributed development which reduces the modeling effort and complexity while improving maintainability and reusability.

In the remainder of this chapter, in Section 5.1, I shortly discuss some considerations with respect to the design guidelines and conditions for the modeling approach and present the modeling approach architecture that has been chosen to tackle these design issues. In this, I briefly justify the design decisions for the architecture and specific modeling concepts, and explain how they tackle particular challenges and requirements identified in Chapter 3. Afterwards, in Sections 5.2 to 5.4, I present the individual components of the modeling approach in more detail by presenting the basic terminology, important definitions, and the key modeling concepts underlying the individual parts of the modeling framework.

5.1 Guidelines, Conditions and Approach

I particularly paid attention to some important design issues during the development of the modeling framework. These are *design guidelines* and *design conditions* that can be considered as meta-requirements with regard to *expressiveness* and *practicability*. They must be considered since the framework is supposed to be used by authors from distinct user groups, in various application areas, with very different demands on the social agents' behavioral capabilities and different requirements for the authors' experience levels and modeling strategies. Taking these design aspects into account, I developed the modeling framework's architecture and modeling concepts that best tackle the challenges described in Chapter 3. I relied on a modular and compositional approach which falls back on specially designed, domain-specific, mainly visual and declarative, modeling languages and descriptive specification languages that are sufficiently expressive but nevertheless highly practicable.

Design Guidelines

EXPRESSIVE &
PRACTICABLE

In general, the design decisions during the development of any domain-specific modeling framework are primarily determined by the intended use, which is essentially defined by the application areas, the user groups, and the methodologies of this framework (van Deursen *et al.*, 2000). Each design usually tries to maximize the *usability* of the modeling approach with regard to these determining factors in order to benefit from the resulting positive effects on, for example, productivity, quality, and acceptance (Hermans *et al.*, 2009; Bariic *et al.*, 2012). The design of a usable modeling approach is often, but at least in this case, characterized and driven by the ubiquitous, alleged opposition between its *expressiveness* and *practicability*. That means, on the one hand, it needs to have quite sufficient expressive power to face the challenges and requirements in the respective application domain. On the other hand, it still needs to be practicable enough, in the sense that it is easily comprehensible and intuitively usable for the intended user groups and allows a quick feasibility from an author's initial, fairly rough idea to a sophisticated, executable, computational model. The gap between expressiveness and practicability is becoming the larger the more complex the application domain or modeling task and the less skilled and experienced the user group.

MODELING
STEPS & STAGES

The development of a behavior and interaction model usually goes through several *modeling stages* during which it is repeatedly drawn up, played through, reconsidered, eventually discussed with others, in parts discarded, and reassembled again, on different *abstraction levels*. This process reflects the constant ascertainment and completion of the authors' ideas and imaginations, and is characterized by the constant refinement of the computational model into a final executable specification. This process initially starts with an often incomplete, rough, and informal sketch of the model, or parts of it, which are frequently drawn or written on paper or a white board. Then, the authors go over to the refinement of the model using more and more formal or semi-formal, but, in any case, more detailed conceptual notations to express the semantics of the model's way of working in more detail until they have finally realized the model in the syntax of an executable specification.

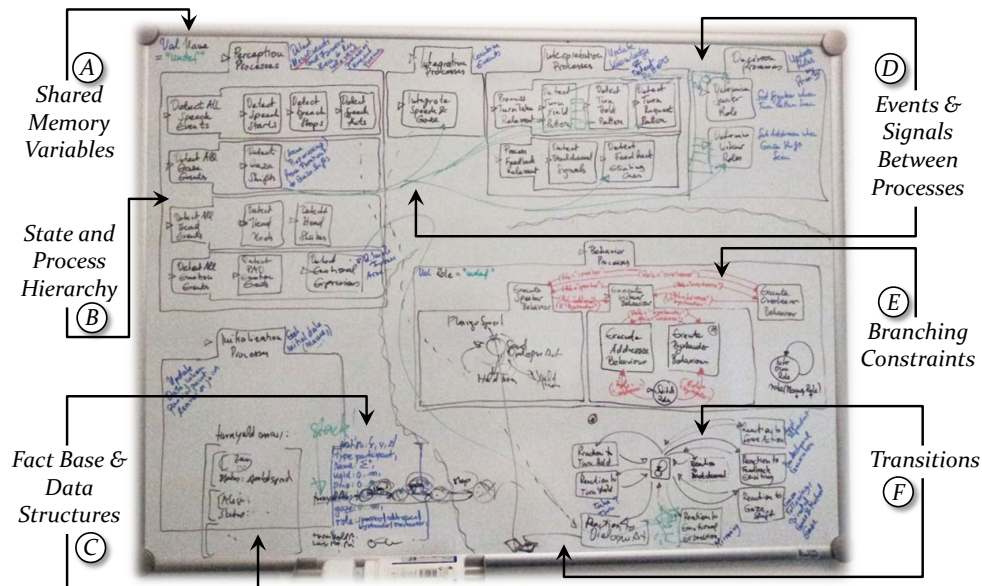


Figure 5.1.1: A white board used to sketch a model with states, transitions, variables and events.

Figure 5.1.1 shows an example of such a white board drawing which was used to draft up and discuss a social agent's behavior in the very early development phase of an interactive application with a physically situated joint activity, similar to the introductory scenario in Section 1.2. Based on these drawings, the draft later developed into the final behavior and interaction model that was realized with the modeling approach developed in this thesis and is illustrated in Chapter 6. Just like in this drawing, people often tend to sketch and discuss their ideas of an interaction or dialog using some kind of state transition diagram consisting of nested and parallel nodes (Figure 5.1.1 (B)) and edges (Figure 5.1.1 (F)). These are then labeled with statements, constraints (Figure 5.1.1 (E)), and behavior descriptions, formulated with keywords or sentences in natural language, or some semi-formal specification format which is usually close to natural language. They use named variables (Figure 5.1.1 (A)) to represent instances and entities in the model and recursive container structures (Figure 5.1.1 (C)) as representation format for more complex data. Finally, they use different kinds of lines and arrows connecting parts of the model that exchange events or signals (Figure 5.1.1 (D)).

Allowing for a quick and intuitive realization of a model from an initial sketch to the final executable specification improves the usability of a modeling approach. Therefore, it is beneficial if the used modeling language and intermediate conceptual representations in the aforementioned development stages can easily be transformed to each other. This can be achieved by ensuring the *syntactic closeness* to each other and, in particular, the initial sketch. Furthermore, many descriptions in the early development phase of a behavior and interaction model are intuitively made in *natural language*. For example, people often formulate integration and fusion constraints or the actions and behaviors that the agent has to perform in natural language, similar to stage directions in a movie script. Therefore, as a special case of syntactic closeness, choosing specification and modeling languages that closely resemble the use of natural language further improves the intuitiveness of the approach.

*SYNTACTIC
CLOSENESS
& NATURAL
LANGUAGE*

Design Conditions

It is now clear that the main design guideline for the modeling approach in this thesis is to achieve practicability by having modeling and specification languages syntactically close to natural language and keeping the intermediate representations from the sketch to the final executable model syntactically close to each other. Besides this general guideline for the design of the modeling approach, some additional design conditions, with regard to the users' experience levels and modeling strategies, have been considered in order to further improve practicability by increasing productivity, effectiveness, and acceptance of the approach.

FACILITATE ITERATIVE DEVELOPMENT In order to achieve high quality and productivity, an author must be allowed to pursue different modeling strategies during the development of a model. On the one hand, he must be able to quickly implement a basic behavior and interaction model in order to test his ideas as well as to try and discard parts of the model. On the other hand, the model must be iteratively refined and qualitatively improved to a more complex and sophisticated behavior and interaction model. In order to develop rather simple models in the initial *rapid prototyping phase*, an author must be able to rely on predefined building-blocks for the description of specific behaviors, such as prefabricated animations or prescribed behavior scripts, as well as input processing functionalities, such as predefined fusion or event signaling and consumption operators. On the other hand, the author must be able to define user-defined constraints, application-specific data structures, and behavior blocks in order to create the more evolved and enhanced models in a later *sophistication phase* of a model's implementation.

COVER LEVELS OF EXPERIENCE The modeling approach must be accessible and usable by experienced and non-expert users. Many related frame-, rule-, or plan-based modeling approaches for behavior and interaction modeling require a substantial degree of expert knowledge and programming skills in order to develop reasonably decent behavior and interaction models. For that reason, they are actually unserviceable for non-computer experts with little programming experience, such as artists and screenwriters, or behavioral psychologists and sociologists, that want to craft interactive applications with virtual characters or social robots. In order to exploit the expert knowledge of these *inexperienced authors* in the area of games, film, theater, and psychology, the modeling technology must be easily accessible and learnable. While these inexperienced user groups might therefore prefer to have a restricted and less complicated set of modeling concepts, the more *experienced authors* might prefer to use much more complex modeling features that could also provide more expressive power as well as more possibilities to configure the sensor and plug-in setup of the system for the target application.

SUPPORT DISTRIBUTED DEVELOPMENT Some behavior and interaction models might become quite complex and difficult to handle for a single author, but, instead, call for the division of the modeling efforts among several shoulders. So, the modeling framework should provide the possibility to model individual parts of the model or approach individual tasks mainly in isolation by distributing them among multiple authors. In this, the individual modeling tasks or aspects can be undertaken by people having the respective expert knowledge, such as psychologists, artists, or computer experts. Exploiting the respective expert knowledge of several authors and dividing

the modeling effort across several shoulders then helps to increase productivity, efficiency, and quality. So, from an author's perspective the modeling framework must allow both, a centralized, that means single-author-oriented modeling approach as well as a distributed, multi-author-oriented modeling method. From the agents' perspective, the authoring tool must support an author-centric approach used for establishing a centralized control for all agents, as well as an agent-centric approach, aiming at modeling agents with autonomy.

Design Approach

Design and conception of the modeling framework in this thesis follow the above described guidelines and conditions. To perform the modeling tasks and meet their task-specific requirements, identified in Chapter 3, I developed the *Behavior Flow Modeling Language (BFML)*. This is an ensemble of domain-specific modeling, scripting, specification and programming languages, each of which fulfills a specific function in the overall behavior and interaction model of a social agent. Through their effective interaction, each of these languages is making a significant contribution to the naturalness and credibility of a social agent's interactive behavior. The individual ensemble members primarily rely on visual and declarative modeling formalisms and textual scripting methods to meet the design guidelines and conditions discussed above. The considerations and decisions leading to their development, the relations among them, and their underlying programming paradigms and modeling concepts are described in more detail in Sections 5.2 to 5.4.

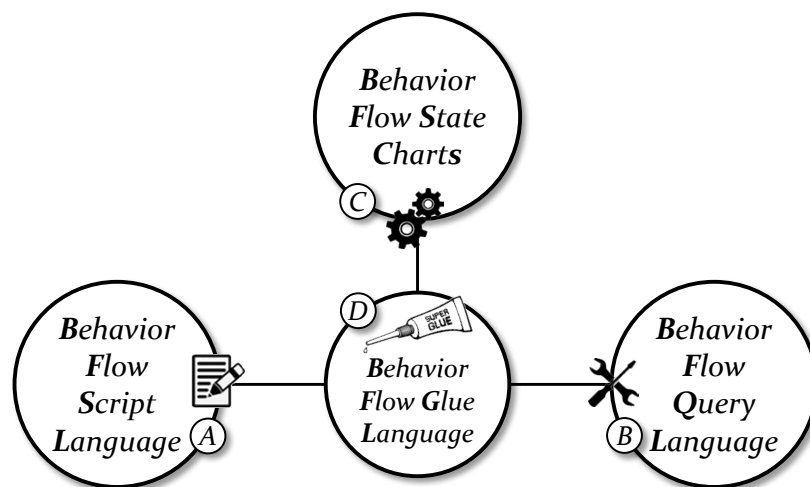


Figure 5.1.2: The architecture of the *BFML* modeling language ensemble developed in this thesis.

Figure 5.1.2 depicts the basic architecture of the *BFML* ensemble with respect to the interrelations and dependencies between the individual languages of the ensemble. The diagram shows that the *BFML* divides the modeling challenge among three rather independent and isolated languages each of which is responsible for one of the individual modeling subtasks identified in Chapter 3. The resulting parts of the model, created with these languages, are combined to create the final computational behavior and interaction model by using a fourth glue language. This modular and compositional nature of the modeling approach has been

chosen because it effectively facilitates the iterative prototyping and the distributed development and, thus, the creation of clearly structured, easily maintainable, and better reusable computational behavior and interaction models:

Behavior Flow Script Language

The *Behavior Flow Script Language (BFSL)* is a declarative, template-based, textual behavior description and specification language used for creating social agents' expressive and credible multi-modal behavior and dialog content (Figure 5.1.2 ©). It can be used for the hybrid creation of knowledge-informed and manually scripted specifications of behavioral activities, such as gaze, gestures, facial expressions, multi-modal utterances, and whole scene performances that resemble parts of a screenplay or movie. With its easy, descriptive syntax, close to natural language, the *BFSL* is very intuitive to use and follows the aforementioned design guidelines and conditions (Gebhard *et al.*, 2003a, 2012).

Behavior Flow Query Language

The *Behavior Flow Query Language (BFQL)* is a largely declarative, and in parts procedural, logic calculus used for processing user input and reasoning on context knowledge (Figure 5.1.2 Ⓓ). It is realized as an embedded, domain-specific language, consisting of facts and rules in *PROLOG* (Kowalski, 1974, 1979). It is mainly used for multi-modal fusion and integration, context and domain knowledge reasoning, and inter-process communication. It fulfills the design guidelines because it allows formulating constraints and operations in first- and higher-order logic (Naish, 1996), which has a long, successful history in the description of syntax and semantics of natural language (Barwise and Cooper, 1981; Pereira, 1983b; Pereira and Shieber, 1987; Montague, 1988).

Behavior Flow State-Charts

The dialog flow as well as behavior and interaction logic are visually modeled with a specially designed, hierarchical and concurrent state-chart variant, called *Behavior Flow State-Charts (BFSCs)* (Figure 5.1.2 Ⓐ). Much of the literature agrees that states, transitions, and events are a priori an intuitive and natural method for describing the dynamic behavior of complex reactive and interactive systems (Harel, 1987), such as the behavior and interaction models of social agents. In addition, state transition diagrams are naturally used to sketch ideas and communicate with each other using drawings, as shown in Figure 5.1.1, while they also underlie many executable specification formats.

Behavior Flow Glue Language

The individual parts of the models, that have been created with the aforementioned languages, are finally connected to a single combined computational behavior and interaction model using an easily comprehensible, imperative, glue language called *Behavior Flow Glue Language (BFGL)* (Figure 5.1.2 Ⓑ). Typically, states and transitions of *BFSCs* contain *BFGL* statements or transition guards that evaluate *BFQL* queries and play back *BFSL* specifications. The *BFGL* is the only language of the ensemble which is not declarative or visual, but, rather resembles a subset of a typical general purpose programming or scripting language (Ousterhout, 1998) and has, as such, been kept simple and focused.

5.2 Creating Behavior and Dialog Content

As identified in Section 3.4, a key modeling task is the creation of expressive and natural multi-modal behavior as well as credible and competent dialog content. The *BFML* meets this task with the domain-specific, declarative and template-based *Behavior Flow Script Language (BFSL)*, a significantly simpler behavior specification method than the well-known, practically standard approaches (Kopp *et al.*, 2006; Vilhjálmsón *et al.*, 2007; Heylen *et al.*, 2008; Kipp *et al.*, 2010; Schröder *et al.*, 2011) which are comparatively rather unhandy and complicated to learn and use for non-experts, such as psychologist, artists, or students. It is a further development of an earlier textual specification format for the multi-modal behavior and dialog content of embodied conversational agents (Gebhard *et al.*, 2003a) that has constantly be extended and adjusted during the course of this thesis. The *BFSL* offers a versatile and intuitive method for the textual specification of variable, expressive, and credible multi-modal behavior and dialog content in an easily understandable rapid prototyping way.

BEHAVIOR
FLOW SCRIPT
LANGUAGE

5.2.1 Behavioral Activity Specification

The *BFSL* uses *behavioral activities* for the flexible, intuitive, and variable textual specification of social agents' multi-modal expressive behavior and dialog content. Its easily readable and comprehensible syntax allows the use of various types and flavors of behavioral activities that differ in terms of the complexity of the behavior they describe, the context in which they are used, and the way how they are scheduled and executed. Behavioral activities in *BFSL* must in no case be confused with behavior specification approaches that solely rely on strictly manually scripted and fully planned ahead content. Instead, they have to be understood as templates that sketch out the structure and operational framework for the description and variation of the agents' behavior. They allow a hybrid specification of multi-modal behavior and dialog content because they can contain both predefined contents and variable parts that can be inferred from application-knowledge and modified at runtime.

BEHAVIORAL
ACTIVITY
SPECIFICATION

Activity Definition

The most basic type of behavioral activity is referred to as *action activity* and is mainly used to specify individual nonverbal behaviors performed by an actor in a single modality, such as simple motor movements, gestures or animations, body postures or movements, single gaze cues or head movements, or facial expressions. In addition, an action activity can be used to define the execution of an application- or device-specific command, for example, an event in an additional graphical user interface or interaction device, such as a surface table, tablet, or smartphone. Finally, it can be used for the execution of commands in the respective agent platform that do not result in a directly observable expressive behavior of the agent, such as, for example, camera movements in the virtual environment of a graphics engine or configuration commands for the text-to-speech synthesizer of a robot platform.

ACTION
ACTIVITY
DEFINITION

Figure 5.2.1 shows the *EBNF* (Backus *et al.*, 1963; Wirth, 1977) syntax diagram of an action activity. An action activity's syntax is easily readable and comprehensible and its meaning is

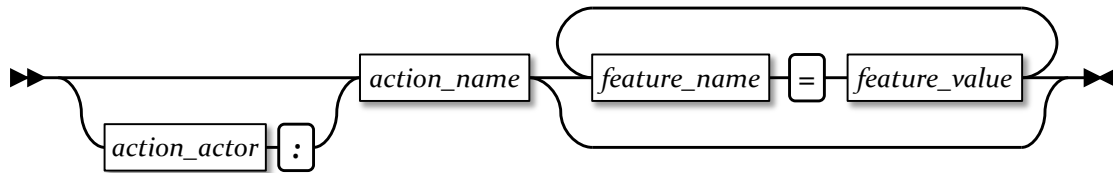


Figure 5.2.1: The syntax of an *action activity* specifying an *actor* and *action* as well as a *feature list*.

usually rather self explanatory. An action activity consists of an *actor identifier* which refers to the unique name of the agent, entity, or device that is supposed to perform the behavior or action. The actor name is always followed by a *colon character* and a mandatory *action identifier* which describes the name of the behavior or modality in which the behavior or action has to be performed. Finally, the actor and action identifiers are followed by an optional list of *action features* which are key-value pairs consisting of a *feature name* and a *feature value*, that are used to parameterize and configure the execution of an action, such as, for example, the number of repetitions or the duration of a behavior, or other action-specific properties. The names of actors and actions as well as the possible feature names and attribute values that are supported by an individual behavioral activity, generally depend on the command interface of the respective virtual character animation engine or motor control unit of the robot platform. Similar to related behavior specification approaches, discussed in Chapter 4, *BFSL* can use a *gesticon* (Pelachaud, 2005) to map action and feature names to the platform-specific commands. The *Interpreter Runtime Environment (IRE)* must then look up to the *gesticon* and replace these identifiers when executing an activity.

- Ⓐ Charly : look target=Marley
- Ⓑ Charly : smile intensity=0.5 duration=2.0
- Ⓒ Charly : tilt direction=right angle=20.0

Figure 5.2.2: Some examples of action activities that specify nonverbal behaviors of the agent Charly.

Figure 5.2.2 shows some exemplary action activities that specify nonverbal behaviors of the agent Charly. The first activity defines a gaze behavior which causes that Charly looks at a specific target named Marley (Figure 5.2.2 Ⓐ). The second one defines a facial expression and determines that Charly has to smile with a moderate intensity for a time period of two seconds (Figure 5.2.2 Ⓑ). The third activity defines a head movement which causes that Charly tilts his head to the right side for an angle of twenty degrees (Figure 5.2.2 Ⓒ).

- Ⓐ Camera : move x=-10.0 y=15.0 z=25.0
- Ⓑ Surface : show id=2 x=120 y=340 w=50 h=100

Figure 5.2.3: Examples of action activities that execute application- and device-specific commands.

Figure 5.2.3 shows some examples of action activities that specify application- and platform specific actions. The first activity represents a command that specifies that the camera has to move to a particular position that is given in tree-dimensional world coordinates of a virtual world (Figure 5.2.3 (A)). The second activity defines a command which causes that an object, for example a photo, with a specific identifier, in this case the number 2, has to be displayed at a particular position and in a certain size on the screen of a surface table (Figure 5.2.3 (B)).

An action activity may be used in two different contexts that impose slightly different syntactical requirements. First, it can be used as a *standalone action activity* if it is used alone, as shown in Figures 5.2.2 and 5.2.3. Second, it can be used as a *nested action activity* when it represents only a part of an enclosing behavioral activity, such as a multi-modal utterance or scene activity. Nested action activities are specified just as their standalone counterparts with the exception that they are additionally enclosed in *square brackets* ([,]) in order to delimit their specification from the enclosing activity. While a standalone action activity always requires an actor, in a nested action activity, the actor name is only an optional argument. If no actor name is specified, then the actor of the enclosing activity, for example, the speaker of an utterance activity or a turn of a scene activity, is implicitly taken by the *IRE* as the actor of the nested action activity. The actor of a nested action activity must not be the same as the one of the enclosing activity. This allows co-verbally aligning one actor's nonverbal behaviors with another agent's spoken words and actions in a multi-modal utterance.

STANDALONE
& NESTED
ACTIVITIES

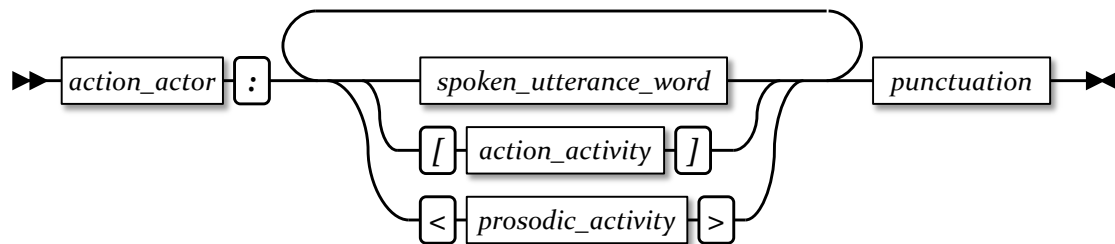


Figure 5.2.4: The syntax of an *utterance activity* with an *actor*, the *content*, and a *punctuation* mark.

More complex behaviors than a simple unimodal behavior or an individual action can be specified with an *utterance activity*. Utterance activities can be used to specify unimodal verbal statements as well as behavioral patterns and multi-modal utterances. As shown in Figure 5.2.4, just like an action activity, an utterance activity consists of a mandatory *actor identifier* followed by a *colon character* and the multi-modal dialog line of the actor as a sequence of *spoken words*, *nested actions*, and *prosodic activities*, which is terminated with a *punctuation mark*. In a multi-modal utterance the interleaving of the individual spoken words with nested action and prosodic activities defines the temporal alignment of the possible actors' verbal statements with their accompanying co-verbal behaviors and actions.

UTTERANCE
ACTIVITY
DEFINITION

Figure 5.2.5 shows some examples of multi-modal utterance activities of the agent Charly containing nested action activities. The first activity specifies that Charly has to say the sentence "Shall I tell you about that?" while, at the same time, he first looks at a photo on the surface table and then, shortly after, raises his eyebrows a bit (Figure 5.2.5 (A)). The second

- Ⓐ Charly : Shall I tell you about [look target=2] that [raisebrows intensity=0.3]?
- Ⓑ Charly : [look target=Marley] Would you like [tilt direction=right angle=20.0] tea?

Figure 5.2.5: Some examples of multi-modal *utterance activities* containing *nested action activities*.

activity defines that Charly asks the question “Would you like tea?” while he simultaneously first looks at Marley and then tilts his head to the right (Figure 5.2.5 Ⓑ).

*NESTED
PROSODIC
ACTIVITIES*

Besides nested action activities, a multi-modal utterance may additionally contain *nested prosodic activities*. An example for such a nested prosodic activity is the *pause activity* which is simply a pause in between two multi-modal utterances that has a certain duration provided in milliseconds. Other prosodic activities can be used to modify the features and tone of speaker’s voice, such as loudness, pitch, timbre, speech rate, and all other *prosodic*, and in parts *paralinguistic*, features that can be customized on the text-to-speech synthesizer of the respective agent or robot platform. A nested prosodic activity does not specify an actor because it refers to the enclosing activity, but, consists of a mandatory action name followed by an optional list of feature-value pairs and is always enclosed with *angle brackets* (<, >).

- Ⓐ Charly : Marley <pause duration=2000> should I get a cup of tea?
- Ⓑ Charly : <volume intensity=1.2> Would you like to have some tea?

Figure 5.2.6: Some examples of multi-modal utterance activities containing nested prosodic activities.

Figure 5.2.6 shows some examples of multi-modal utterance activities of the agent Charly containing nested prosodic activities. The first utterance activity specifies that the agent has to say “Marley ...” followed by a pause of two seconds before continuing with “... should I get a cup of tea?” (Figure 5.2.6 Ⓐ). The second utterance activity defines that the agent says the sentence “Would you like to have some tea?” with a louder voice than usual (Figure 5.2.6 Ⓑ).

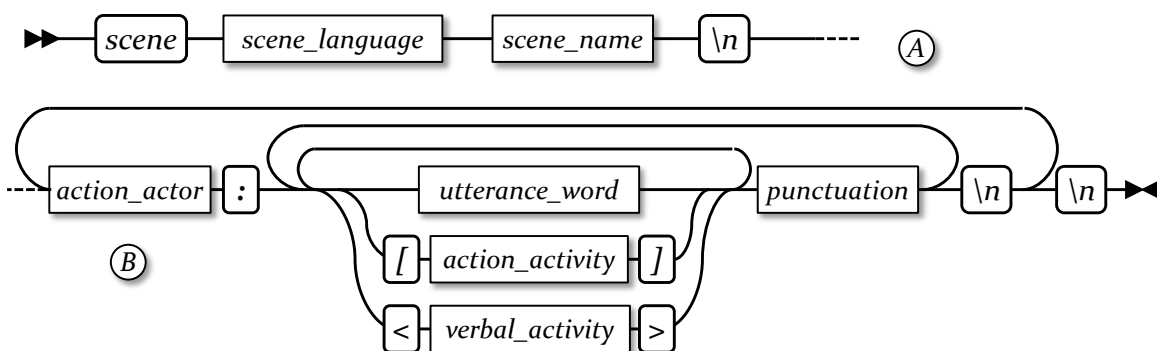


Figure 5.2.7: The syntax of a *scene activity* with the *scene header* (Ⓐ) and the *scene body* (Ⓑ).

*SCENE
ACTIVITY
DEFINITION*

The most complex type of behavioral activity is a *scene activity* which can be used for the specification of whole interactive performances consisting of sequences of multi-modal turns

and utterances of multiple agents. A scene thus specifies the interleaving of the individual agents' verbal statements and stage directions for the co-verbal alignment of their nonverbal behaviors and platform-specific commands, such as gestures, postures, gaze behaviors, and facial expressions. Scenes are specified in an external textual *scene script* which resembles a part of a screenplay or movie script. As shown in Figure 5.2.7, a scene specification is composed of a *scene header* that must be terminated with a newline and a *scene body* that has to be separated with a blank line from the following scene definitions in the scene script. A scene header always starts with the initial keyword *scene* which marks the definition of a new scene and is followed by the specification of the scene's language with a valid *language identifier*, for example, *en* for English or *de* for German. The language specification is followed by a single identifier that defines the name of the scene followed by the terminating newline. The body of the scene then contains a sequence of the individual agents' turns each of which is itself composed of a sequence of the respective agent's multi-modal utterances. These multi-modal utterances are defined just as utterance activities, that means, they contain the spoken text of the respective agent interleaved with nested verbal and action activities.

- Ⓐ *scene en welcome*
 Charly : Hello [Reeti : look target=Marley] Marley!
 Ⓑ Reeti : Yes, hi Marley <pause 2000> tell us, how are you today?

Figure 5.2.8: An example for the specification of a scene activity with the two agents Charly and Reeti.

Figure 5.2.8 shows an exemplary specification of a scene that defines the behavior of the agents Charly and Reeti during an interaction with the user Marley. The header of the scene specifies that the language of the scene is English by using the language identifier *en* and that the name of the scene is *welcome* (Figure 5.2.8 Ⓐ). The body of the scene specifies the behavior of the two agents Charly and Reeti in a dialog during which they welcome the user Marley (Figure 5.2.8 Ⓑ). In the first turn, Charly says the sentence “Hello Marley!” while Reeti starts looking at Marley. The second turn actually consists of four separate utterances of Reeti, first “Yes, ...”, then “... hi Marley”, followed by a pause of two seconds, then “... tell us, ...” and, finally, “... how are you today?”.

Execution Policies

Behavioral activities are played back in separate *activity worker threads* of the *IRE* when calling built-in *BFGL* commands from the *BFSC*. In this, a behavioral activity can be executed by the *IRE*'s calling state-chart process with two different *activity execution policies*. The *blocking policy* causes that the calling process blocks in the *BFGL* statement until the activity worker thread has been terminated either by regularly finishing or being interrupted and preemptively aborted. In order to realize the blocking policy, the animation engine or motor control of the agent or robot platform has to acknowledge the execution of animations, motor movements, and speech synthesizing procedures via appropriate notification events of a suitable communication protocol. On the other hand, an activity may be executed with

ACTIVITY
EXECUTION
POLICIES

a *non-blocking policy* such that the calling *IRE* process continues with the execution of the next statements or outgoing transitions of the node while the behavioral activity is *concurrently* executed in the activity worker thread. This “*fire-and-forget*” method does not require any monitoring or notification protocol between the *IRE* and the agent or robot platform.

ACTIVITY PLAYBACK COMMANDS The *BFGL* provides different built-in playback commands for behavioral activities that syntactically vary for the blocking and non-blocking execution policy. The syntax of the playback commands includes an exclamation mark (!) to reflect the imperative character of the statement that is used to instruct one or more virtual characters or robots to perform a behavior or action. An activity playback command always starts with an opening !- for the blocking, or != for the non-blocking policy, and ends in both cases with a closing point (.).

- Ⓐ != Charly : nod repetitions=2 extent=1.0 .
- Ⓑ !- Charly : point target=2 duration=2.0 .
- Ⓒ !- Charly : Shall I tell you about
[look target=2] that [raisebrows intensity=0.3]? .

Figure 5.2.9: Some examples of *blocking* and *non-blocking* playback calls for different activities.

Figure 5.2.9 shows some examples of blocking and non-blocking playback command calls for different standalone action and utterance activities of the agent Charly. The first command calls a non-blocking action activity which causes that Charly nods with his head two times (Figure 5.2.9 Ⓐ). The second command calls a blocking action activity that lets Charly point to a specific photo for two seconds (Figure 5.2.9 Ⓑ). The last command calls a blocking utterance activity that causes that Charly asks the question “*Shall I tell you about that?*” while looking at an object and raising his eyebrows (Figure 5.2.9 Ⓒ).

SCENE ACTIVITY EXECUTION In contrast to the playback of action and utterance activities, which are specified *inline*, that means, directly in the playback command statement of the *BFGL* within the *BFSC*, a *scene activity* is called by referring to the name of a *scene* from an external *scene script*. The scene script is structured like a screen or movie script and contains a sequence of scenes some of which may even have the same identifier. Figure 5.2.10 shows an example of a scene activity playback call to the scene shown in Figure 5.2.8. The scene activity playback statement contains the name of the scene enclosed by an opening !- and a closing point (.).

!- welcome .

Figure 5.2.10: An example of an activity playback command that executes the scene from Figure 5.2.8.

NESTED ACTIVITY EXECUTION The verbal part of a multi-modal utterance, whether within an utterance activity or a turn of a scene activity, must always be performed by the agent as a single, fluent, and coherent verbal statement without any pauses. For that reason, all nested action activities within such a multi-modal utterance are automatically executed with a *non-blocking policy*. Consequently,

the thereby generated co-verbal behaviors and actions are always performed *in parallel* to the spoken utterance without interrupting the text-to-speech synthesis process. However, this might lead to situations in which the resulting animations and commands are temporary overlapping or even conflicting because they require the same resources, such as motors of a robot's limbs or animation targets of a virtual character. It is up to the animation engine of the virtual character or the motor control of the robot platform to handle those overlaps and resolve possible conflicts, for example, by aborting a behavior in favor of another or by blending and mixing behaviors with each other. Nested prosodic activities, in contrast, are always executed with a *blocking policy* because they need to have an *immediate effect* on the speech synthesis. In particular, pauses split the enclosing verbal statement into two separate parts that have to be individually synthesized by the text-to-speech system of the agent or robot platform, one after the other. While the nested activities are non-blocking in the scope of the enclosing utterance activity, they nevertheless belong to it such that the execution of a multi-modal utterance as a whole is only finished as soon as all therein contained nested activities have been terminated and the whole spoken part has been fully synthesized.

The *BFGL* finally provides a *behavior abortion command* to realize the preemptive abortion of an agent's behaviors and actions. In order to achieve this abortion, the agent platform must be able to stop all motor movements, animations, and text-to-speech synthesis procedures ahead of schedule. The resulting forced notifications of the agent platform then release all blocking playback calls in the *BFSC* that execute activities which exclusively contain behavior specifications of this agent. As shown in Figure 5.2.11, the abortion statement always starts with an opening !~ followed by the agent's name and closes with a point (.).

ACTIVITY
EXECUTION
ABORTION

!~ Charly .

Figure 5.2.11: The command that is called to achieve the abortion of an agent's current behavior.

5.2.2 Parameterization and Variation

With the *BFSL*, the challenge of the flexible integration of knowledge into the behavior and dialog content of a social agent is faced with two different mechanisms. On the one hand, an author may use an *inline value insertion* method by referring to variables of the *BFSC* directly within the specification of action and utterance activities. On the other hand, a scene activity playback command may be extended with a list of *substitution arguments* for specific *placeholder variables* in a scene specification. Both methods represent an easily comprehensible way to integrate context and application-specific knowledge into the contents of an agent's actions, behaviors, and dialog lines. This method of manual insertion of variable content is more flexible but also less generative and automated than approaches pursued by some related behavior specification methods (Cassell *et al.*, 2004; Carolis *et al.*, 2004).

Placeholder Variables

PLACEHOLDER
VARIABLE
SUBSTITUTION

A scene specification may contain special *placeholder variables* which can be replaced with values that are handed over as arguments of a scene activity playback command, called from the *BFSC*. Such placeholder variables are usually used to parameterize parts of an agent's verbal statements, for example, individual words of the spoken utterance, or feature values, for example, the position of a gaze target, but can also be used to vary the name of the actor, action, or a feature. The values that are used for their substitution may be literals of primitive types, such as integral or floating point numbers or string values, but are more often values of state-chart variables that hold context knowledge retrieved in beforehand, for example, using a *BFQL* query to the *PROLOG* fact base or calling a function on a plug-in module.

HYBRID
CONTENT
DEFINITION

The possibility to parameterize parts of a scene activity allows creating behavior specifications in a hybrid way between fixed, authored content and variable, generated content. The behavior specified by the scene activity may, of course, still be manually scripted at large for the purpose of rapid prototyping, however, it can additionally be enriched, in parts, with automatically generated content that has been generated by external reasoning modules from context and domain knowledge and stored in the variables of the *BFSC*. This mechanism can significantly contribute to the variation and competence of behavior and dialog content and, thus, to a more vivid, variable, credible, and less repetitive behavior of the social agents.

scene en welcome

- Ⓐ *Charly : Hello [Reeti : look target=\$user] \$user!*
Reeti : Yes, hi \$user <pause \$time> tell us, how are you today?
- Ⓑ *!- welcome user=UserName, time=2000 .*

Figure 5.2.12: A scene definition with *placeholders* (Ⓐ) and a call to this scene with *arguments* (Ⓑ).

Figure 5.2.12 shows an example of a scene that contains different placeholder variables (Figure 5.2.12 Ⓐ) which are used to enrich the manually scripted content of the scene with variable content handed over from the calling *IRE* process as arguments of the playback command (Figure 5.2.12 Ⓑ). The scene has basically the same function as the scene shown in Figure 5.2.8 with the exception that the name of the user and the duration of the pause can be variable. Therefore, the scene uses the placeholder variable *\$user* as gaze target in the nested action as well as name of the user in the sentences of the two agents. In addition, the scene uses the variable *\$time* as duration of the speech pause of the agent Reeti. The substitutions for the placeholder variables are handed over as arguments of the playback command in form of feature value list following the identifier of the scene. In the above example, the placeholder variable *\$user* is replaced with the value of the state-chart variable *UserName* and the variable *\$time* is replaced with the literal integer value 2000 (Figure 5.2.12 Ⓑ).

Inline Value Insertion

Besides the use of the aforementioned *placeholder variables* in scene specifications, another method for the integration of domain and context knowledge into the agents' behavior and dialog content is the *inline value insertion*. This method directly uses the names of state-chart variables in the *BFGL* playback statements for an action or utterance activity. The *IRE* then replaces the variable names with their current values whenever executing such a playback command in a node of the *BFSC*.

- Ⓐ !- Charly : point target=*Target*↓ duration=2.0 .
 Ⓑ !- Charly : This was your trip through *Destination*↓ in *TravelDate*↓! .

Figure 5.2.13: Some examples of *inline value insertion* when calling action and utterance activities.

Figure 5.2.13 shows some examples of the inline insertion of values when calling action and utterance activities. The first command calls an action activity that causes the agent named Charly to point to a certain object on the surface table, whose name is represented by the value of the state-chart variable *Target*, for a duration of two seconds (Figure 5.2.13 Ⓐ). The second command calls an utterance activity that causes that the agent says a sentence in which he refers to a travel destination and date given by the values of the state-chart variables *Destination* and *TravelDate*, respectively (Figure 5.2.13 Ⓑ). For example, if the destination would be *France* and the date would be *1980*, then the agent would say the sentence “*This was your trip through France in 1980!*” like in scene ③ of the introductory example from Section 1.2.

Grouping and Blacklisting

The challenge of the *automatic variation* of behavior and dialog content can already be tackled to some degree with the template-based specification approach with behavioral activities. As mentioned before, both placeholder variables and inline value insertion can be used to enrich the behavior and dialog content of the social agents with variable content that can be inferred from domain and context knowledge. Another very efficient and intuitive method for the variation of content at runtime can be taken advantage of by defining *scene groups* within a *scene script*. Scene groups are created by providing a number of variations for each individual scene when writing a scene script. Multiple scenes that share the same name and language identifiers are then grouped together and organized to a scene group which has the same name as the contained scenes. The creation of scene groups then enables the automatic variation of content at runtime by using a particular *blacklisting strategy* for their execution. Each time when calling a scene activity from such a scene group, then, for example, a *randomized* or *linear blacklisting mechanism* can be applied by the *IRE* to determine one particular alternative from the existing variants of the scene for execution. After the scene has been executed, it is pushed to the blacklist, such that it is not executed again until all other scenes of the same group, that have not yet been blacklisted, are executed before. This method can

effectively be used to avoid repetitive behavior in successive calls to the same scene activity which makes the agent’s behavior appear much more variable and natural. The default blacklisting strategy has to be configured as part of the project configuration in the *Behavior Flow Configuration Language (BFCL)* and may additionally be set using corresponding methods of the *BFGL*. Figure 5.2.14 shows an example of a scene group consisting of two different variations (Figure 5.2.14 (A), (B)) of a scene with the same name.

```

scene en welcome
(A) Charly : Hello [Reeti : look target=$user] $user!
    Reeti : Yes, hi $user, how are you today?

scene en welcome
(B) Reeti : Welcome [look target=$user] $user, good to have you here!
    Charly : Oh, see who is here! Hello $user, how are you today?

```

Figure 5.2.14: A scene group with two different variations ((A), (B)) of a scene with the same name.

5.3 Integrating Input and Context Events

BEHAVIOR
FLOW QUERY
LANGUAGE

As explained in Section 3.3, an important modeling task is the proper processing and understanding of multi-modal inputs and context events including the reasoning on context knowledge. The *BFML* meets this task with the *Behavior Flow Query Language (BFQL)*, a mainly declarative, domain-specific, logic calculus that is implemented and embedded in *PROLOG*. It has been developed in this thesis as completely novel extension of an earlier, exclusively state-chart-based, behavior and interaction modeling approach (Gebhard *et al.*, 2003a) that was lacking the possibility to manage complex knowledge and event structures for multi-modal fusion and knowledge reasoning. Predefined first- and higher-order logic predicates are used for temporal reasoning and quantification at multi-modal fusion and may be complemented with application-specific predicates for knowledge reasoning and semantic constraints (Mehlmann and André, 2012; Mehlmann *et al.*, 2016). Its design is inspired by rule-based (Hoste *et al.*, 2011; Dumas *et al.*, 2014), plan-based (Rich and Sidner, 1998; Bohus and Rudnicky, 2003) and complex event processing (Luckham, 2001; Anicic *et al.*, 2010, 2011; Bruns and Dunkel, 2015) methods. It is influenced by multi-modal fusion methods based on feature structure unification (Cohen *et al.*, 1997a; Johnston *et al.*, 1997), multi-modal grammars (Johnston, 1998b, 2000), finite-state automata (Johnston and Bangalore, 2001, 2005; Bangalore and Johnston, 2009) and state transition networks (Latoschik, 2002, 2005).

5.3.1 Feature Structure Representation

THE FEATURE
STRUCTURE
FORMAT

The knowledge about the application domain and context information as well as the users’ input events in the various information modalities is represented with a data structure known as *feature structure* (Carpenter, 1992; Pereira, 1993). Generally, all kinds of feature structures are hierarchical data structures that are used for representing partial information about an

object or an event that is expressed in terms of *features* or attributes and their corresponding *values*. Being a well-formed *uniform representation format* they are ideally suited to carry arbitrary information from different levels of abstraction and processing stages and may contain data ranging from purely lexical or symbolic information, such as gaze coordinates, to high-level semantic interpretations, such as dialog acts or referenced objects of a pointing gesture. This makes them a helpful method for handling the heterogeneity and irregularity of the information carried by the multi-modal input and context events and contained in the domain knowledge. They are easily adaptable to application- and device-specific properties which helps to overcome compatibility and extensibility issues. In this, they do not suffer from syntactic overhead and are less restrictive or application-specific than more specialized formats (Wasinger *et al.*, 2005; Wasinger, 2006; Johnston, 2009).

Feature structures have a long history in a broad range of research areas that investigate problems which are related to the issues approached in this thesis. For example, they have been used in *unification-based* formalisms for natural language parsing and understanding in the area of computational linguistics (Shieber, 2003). Those include basic phrase structure (Kay, 1979, 1984; Shieber *et al.*, 1983; Shieber, 1984, 1985) as well as equation-based (Kaplan and Bresnan, 1982) and constraint-based formalisms (Pollard and Sag, 1987, 1994). Related *frame-based* description formats are used in various knowledge representation and automated reasoning formalisms (Minski, 1975; Fikes and Kehler, 1985; Brachman *et al.*, 1983; Brachman and Levesque, 2004). Related *record-based* representation formats are employed in logic programming languages for constraint logic programming and constraint-based unification grammar formalisms (Carpenter *et al.*, 1991; Smolka, 1992; Ait-Kaci *et al.*, 1994; Smolka and Treinen, 1994; Backofen and Smolka, 1995). As shown in Chapter 4, feature structures and similar formats have been found to be useful in the domain of unification-based multi-modal fusion and parsing (Cohen *et al.*, 1997b; Johnston *et al.*, 1997; Johnston, 1998b; Wu *et al.*, 1999; Oviatt *et al.*, 2000; Alexandersson and Becker, 2003; Pflieger, 2004; Portillo *et al.*, 2006; Sun *et al.*, 2007; Oviatt, 2012; Ehlen and Johnston, 2013; Kaiser *et al.*, 2003; Holzapfel *et al.*, 2004; Stiefelhagen *et al.*, 2004; André *et al.*, 2014; Mehlmann and André, 2012; Mehlmann *et al.*, 2014a, 2016).

FEATURE
STRUCTURE
ORIGINS

The term feature structure is not an unambiguously defined concept but rather refers to a whole family of similar representation formats in the area of feature description languages. These flavors of feature structures are in parts syntactically similar looking and semantically close to each other, or even equivalent, but, differ in their use and the research area from which they originate. The formal, underlying definition of feature structures used in this thesis is a slightly adapted version of previous definitions that consider feature structures as a kind of *labeled finite-state-automaton* (Kasper and Rounds, 1986, 1990; Carpenter, 1993) that can be represented as *directed acyclic graphs*. Our, in this sense, *graph-theoretic definition* of feature structures is however slightly simplified due to the reason that it does not require feature structures to have the ability of *structure sharing* (Pollard and Sag, 1994).

FEATURE
STRUCTURE
DEFINITION

Feature structures can be depicted using two different notation formats. A feature structure can graphically be represented as *acyclic directed graph* whose edges are labeled with fea-

FEATURE
STRUCTURE
NOTATIONS

ture names, inner nodes represent nested feature structures and leaf nodes are labeled with feature values. This graphical notation is oriented towards their formal, graph-theoretic definition and well-suited for designing and illustrating formal definitions of operations on feature structures by exploiting algorithms from graph theory. However, the graphical notation as labeled state transition diagram can become cumbersome to read and unwieldy for large feature structures and requires rather complex data structures for the implementation. For this reason, a feature structure can also be represented using a textual specification format as nested *attribute-value matrix*. In this notation, each bracketed grouping corresponds to an inner node while each attribute-value pair corresponds to a leaf node in the graphical representation. The *empty feature structure* has the notation $[]$ in the matrix notation and \top in graph notation. The matrix notation is better suited to encode feature structures as closed terms, for example, for the serialization in general purpose programming languages, such as *PROLOG*. Whether in graph or matrix notation, for the remainder of this thesis, in both notations, feature names are displayed in small capitals, such as `TYPE` or `DATA`, and feature values are displayed in italic face, such as *X*, *square* or *'Hello World!'*, while variables always begin with an upper-case letter, such as *X* or *Name*.

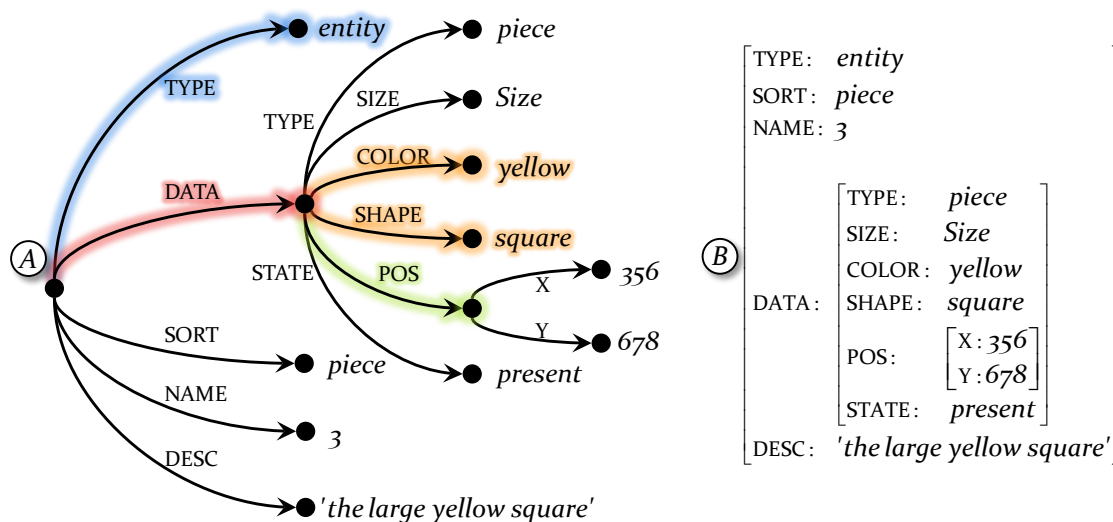


Figure 5.3.1: The representation of a feature structure in graph-notation (A) and matrix-notation (B).

Figure 5.3.1 shows an example of a simple feature structure that is depicted in its notation as directed acyclic graph (Figure 5.3.1 (A)) as well as in the attribute-value matrix notation (Figure 5.3.1 (B)). The feature structure has, for example, a feature with name `TYPE` and value *entity*, highlighted bluish in the graph, and contains a feature with name `DATA` that points to a nested feature structure, highlighted reddish in the graph. This nested feature structure has again has a second nested feature structure at the feature with name `POS`, highlighted greenish in the graph, and, for example, a feature with name `COLOR` and value *yellow* and a feature with name `SHAPE` and value *square*, both highlighted orange in the graph.

FEATURE STRUCTURE ENCODING A variety of research has shown that feature structures can be encoded as different kinds of *PROLOG* terms (Mellish and Gazdar, 1989; Schöter and Place, 1993; Gerdemann, 1995; Cov-

ington, 1994). Consequently, in the *BFQL*, feature structures are represented, based on the attribute-value matrix notation, in form of closed, recursive list terms that contain nested *PROLOG* terms for the representation of the individual feature-value pairs. The *BFQL* encoding of feature structures as simple *PROLOG* lists is inspired by similar approaches that have been using open-ended list terms in which each element corresponds to a feature-value pair (Pollard and Sag, 1987; Mellish and Gazdar, 1989). The feature structure encoding used by the *BFQL* currently relies on closed lists because, even though falling back on *PROLOG* unification too, the current implementations of the *BFQL*'s basic feature structure operations rely on a non-destructive approach and realize different functions than simply a destructive unification as done by the approaches using open lists. A feature-value pair in *BFQL* is represented as a binary functor term with the colon symbol as functor name (:), the feature name as first and the feature value as second argument and is usually used in its infix operator form ($f : v$). Feature-value pairs are separated with comma symbols (,) within a list which is framed with an opening square bracket ([) and a closing square bracket (]). For example, Listing 5.3.1 shows the *PROLOG* encoding of the feature structure shown in Figure 5.3.1.

```
[ type : entity,
  sort : piece,
  name : 3,
  data : [ type : piece,
          size : Size,
          color : yellow,
          shape : square,
          pos : [ x : 356, y : 678 ],
          state : present ],
  desc : 'the large yellow square' ]
```

Listing 5.3.1: The *BFQL* encoding of the feature structure from Figure 5.3.1 as list-based *PROLOG* term.

5.3.2 Logic Fact Base and Event History

The *BFQL* is used to manage a well-organized working memory containing logic facts (Kowalski, 1974; Emden and Kowalski, 1976; Kowalski, 1979) representing domain and context knowledge (Brachman and Levesque, 2004) as well as input events in form of feature structures which are encoded as closed-list terms in *PROLOG* (Pollard and Sag, 1987; Mellish and Gazdar, 1989). It defines dynamic predicates for the wrapping, assertion, retraction, and modification of feature structures in the fact base. The back-bone of the *BFQL* is made-up by a few logic and procedural predicates that are used for basic inference, matching, and manipulation operations on feature structures, such as the retrieval or comparison of feature values, paths and, substructures as well as the construction and modification of feature structures.

Feature Structure Operations

Based on the formal, graph-theoretic definition of feature structures and their underlying concepts, a variety of basic operations for the retrieval, matching, and modification of a feature structure's contents were defined. These operations have then been implemented in

*MATCHING &
RETRIEVAL*

SWI-PROLOG (Wielemaker *et al.*, 2012) and represent an essential part and backbone of the *BFQL*. Due to lack of space, presenting the definition and implementation of each of these operations goes beyond the scope of this thesis. Listing 5.3.2 exemplarily shows the reference implementation of the, probably most important feature structure operation *val/3*. The implementation of the other predicates can be found in the *BFQL* source code that comes with the open-source version of the *VSM*³ authoring software¹ developed in this thesis.

```
val(Feature, Value, [Feature:Value_]) :-
    fkeyterm(Feature). /* Value for a simple feature name */
val(Feature:Path, Value, [Feature:Record_]) :-
    \+allvarls([Feature, Path, Record]),
    val(Path, Value, Record). /* Recursion into nested record */
val(Feature, Value, [_Record]) :-
    nonvar(Record),
    val(Feature, Value, Record). /* Iteration over this record */
```

Listing 5.3.2: The *SWI-PROLOG* implementation of the basic feature structure predicate *val/3*.

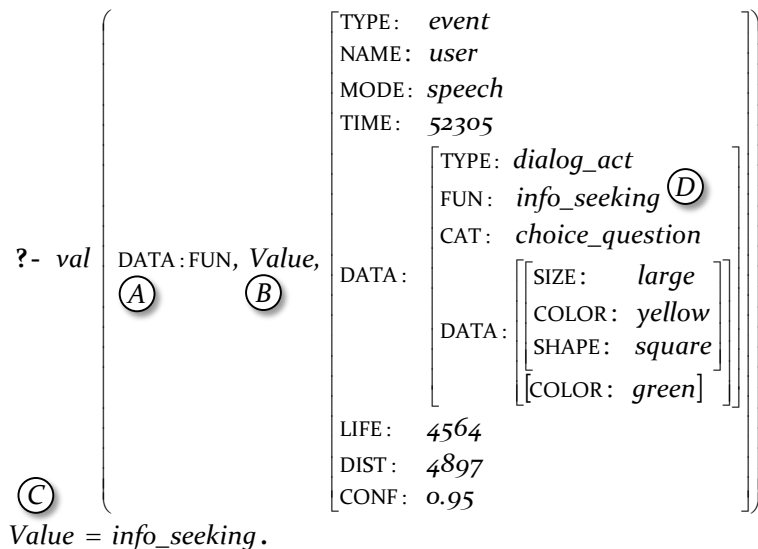


Figure 5.3.2: An example for the use of the logic matching operation predicate *val/3* in a query.

Figure 5.3.2 shows an exemplary application of the predicate *val/3* in a query to the *PROLOG* fact base. In this case, it is used to retrieve the value at path `DATA:FUN` (Figure 5.3.2 A) of the given feature structure (Figure 5.3.2 D) and unify the found value with the variable *Value* (Figure 5.3.2 B) in the case of the successful evaluation of the query (Figure 5.3.2 C).

ATTRIBUTE MANIPULATION The *BFQL* includes various additional basic operation predicates on feature structures. For example, the insertion predicate *add/4* inserts a certain value or substructure at a specific path of an input feature structure and produces an output feature structure containing the new feature-value pair. The removal operation *del/3* removes a value or substructure at a specific path of a given input feature structure and returns the modified output feature structure.

¹<http://scenemaker.dfki.de>

The modification predicate *set/4* modifies an input feature structure by setting a new value or substructure at a certain path of the structure. It can also be used to check if two feature structures differ or can be transferred into one another via the modification of a single feature-value pair.

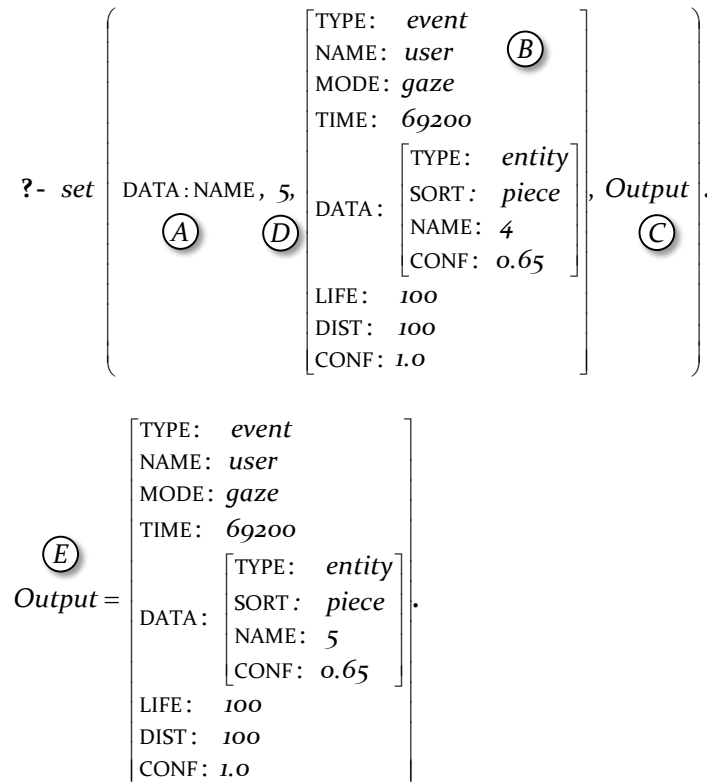


Figure 5.3.3: An example for the use of the logic modification operation predicate *set/4*.

Figure 5.3.3 shows an exemplary usage of the modification predicate *set/4* in a query to the *PROLOG* fact base. At successful execution (Figure 5.3.3 (E)), it can replace an already existing value or structure at the feature path *DATA:NAME* (Figure 5.3.3 (A)) in the given input structure (Figure 5.3.3 (B)) with the new value 5. If no such value exists, but a sub-structure at path *DATA*, then it creates a new feature at path *DATA:NAME* and initializes it with the value 5 (Figure 5.3.3 (D)). Finally, it unifies the variable *Output* (Figure 5.3.3 (C)) with the feature structure that results from this replacement or insertion (Figure 5.3.3 (E)) on the input structure.

Event History Management

Besides facts, representing knowledge about the task, users, agents, and objects in the environment, feature structures are used to represent input events carrying information about the users' behaviors in the various modalities. This includes input devices such as microphones, eye-trackers, and cameras, but also interaction devices such as tablet computers and surface tables. Their raw data is preprocessed by modality-specific interpretation modules, which are usually synchronized, for example, using the *SSI* framework (Wagner *et al.*, 2013), and the thereby produced multi-modal events are afterwards asserted to the *PROLOG* fact base.

*INPUT EVENT
PROPERTIES*

Passing through this preprocessing and synchronization pipeline, an event is equipped with general features, such as the name of the user, the inducing modality or device, recognizer-specific confidence values as well as timestamps of the underlying behaviors' occurrence and duration and the event's assertion time. In addition, they carry modality-specific semantic information provided by the respective interpretation module, such as gaze targets, recognized gestures, or facial expressions as well as dialog acts parsed from speech transcripts.

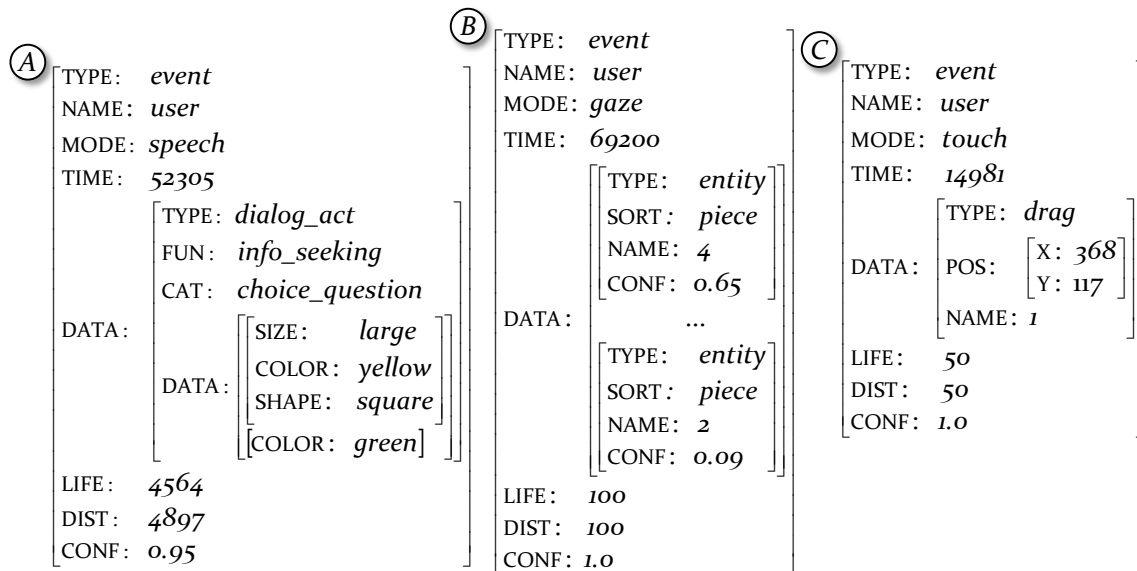


Figure 5.3.4: Exemplary feature structures that are representing input events produced by the user.

Figure 5.3.4 shows some exemplary feature structures representing different input events. The structure shown in Figure 5.3.4 (A) represents the meaning of the user's choice question "do you mean the large yellow square or the green thing?" in form of an abstract dialog act. Its semantic content comprises the communicative function and a semantic category as well as application-specific knowledge, such as the colors, sizes, and shapes of the referenced objects. The structure shown in Figure 5.3.4 (B) represents a gaze event containing a distribution of probabilities with which the user looks at certain objects in the environment. The structure shown in Figure 5.3.4 (C) represents a touch event comprising the type of the touch interaction, in this case a dragging action, the name of the object which has been manipulated or moved and the location of the touch event as two-dimensional coordinates on the touch screen.

WELL-FORMED INPUT EVENTS

Just as those shown in Figure 5.3.4, all features structures representing multi-modal input events need to meet specific content-wise and structural requirements in order to guarantee the consistency of the event history and the proper functioning of the *BFQL* predicates. The different time stamps, such as the time since an event's occurrence, its total lifetime duration and its arrival time, are essential for reasoning about the temporal relations between events using the corresponding *BFQL* predicates. In order to reasonably combine the partial semantic information of different events they not only need to carry a semantic content with different hypotheses but also the confidence values expressing the probability of each of

these hypotheses. Therefore, without limiting the generality, in the remainder of this thesis we demand that a multi-modal input event must possess this basic set of such indispensable features. Events satisfying this restriction are *well-formed events* and each finite set of well-formed events asserted to the fact base is a *well-formed event history*. Of course, the level of detail in the representation of an input event provided by this restriction can easily be adapted to existing standards (Johnston, 2009) or to application specific requirements.

The event history is realized as part of the *PROLOG* fact base and contains well-formed events asserted as special facts using the dynamic *BFQL* predicate *fsr/1*. The *BFQL* provides a number of additional event history management predicates that can be used to assert and retract individual feature structures or a specific set of feature structures that have particular properties. For example, Listing 5.3.3 shows the *SWI-PROLOG* implementation of the *BFQL* predicates *del/1*, *add/1* and *rll/2* for the assertion and retraction of event feature structures

EVENT HISTORY
MANIPULATION

```
del(Record) :- retract(fsr(Record)).
add(Record) :- assertz(fsr(Record)).

rll(Path, Value) :-
    forall((fsr(Record), val(Path, Value, Record)), del(Record)).
```

Listing 5.3.3: The *SWI-PROLOG* implementation of the basic *BFQL* predicates *del/1*, *add/1* and *rll/2*.

Events that are probably not required for multi-modal fusion or reasoning any more need to be retracted from the event history. This helps to keep the fact base reasonably small and thus the logic inference and backtracking mechanism sufficiently real-time capable. For that purpose, the *BFQL* provides predicates that can be used to realize a scalable, age-based, modality-specific *garbage collection* mechanism. They can regularly be called in dedicated processes of a *BFSC* to retract events of a certain modality from the event history whenever they have reached a certain age. Since these events cannot be considered by the inference mechanism anymore, the decision how long to preserve them must carefully be based on the different modalities' mean processing delays as well as empirical data on timing and alignment constrains in multi-modal human interaction. Listing 5.3.4 shows the *SWI-PROLOG* implementation of the exemplary *BFQL* garbage collection predicates *clean/2* which is used to retract all events of a certain modality that have reached a particular age.

GARBAGE
COLLECTION

```
clean(Mode, Age) :-
    now(Now), Lim is Now - Age,
    forall((fsr(Record),
            val(mode, Mode, Record), val(time, Time, Record),
            val(dist, Dist, Record), val(life, Life, Record),
            End is Time - Dist + Life, Lim > End),
           retract(fsr(Record))).
```

Listing 5.3.4: The *SWI-PROLOG* implementation of the *BFQL* garbage collection predicate *clean/2*.

5.3.3 Multi-Modal Fusion and Reasoning

Falling back on the aforementioned fact base management and feature structure operations, the *BFQL* defines predicates that allow answering diverse queries concerning the condition and composition of the event history, such as the existence of events with specific attributes or the comparison of two events' properties. Other predicates are used for the computation of delays and timeouts or the production and consumption of signals for inter-process communication and data exchange via the *PROLOG* fact base. Finally, it provides first-order logic predicates to evaluate quantitative and qualitative temporal and ordering relations between events as well as higher-order meta-predicates that implement generalized quantifiers, which are required for multi-modal event fusion. This section presents the *SWI-PROLOG* implementations of a very restricted selection of predefined *BFQL* predicates.

Basic Predefined Predicates

*RETRIEVAL &
COMPARISON*

Based on the definitions of well-formed events and the basic feature structure operations, the *BFQL* defines a variety of predicates for inspecting, retrieving, and comparing individual attributes of feature structures. Listing 5.3.5 shows the *SWI-PROLOG* implementation of the exemplary retrieval predicate *mode/2* and the comparison predicate *equal_mode/2*.

```
mode(Event, Mode) :-
    fsr(Event), val(mode, Mode, Event).

equal_mode(Event1, Event2) :-
    mode(Event1, Mode1), mode(Event2, Mode2), Mode1 == Mode2.
```

Listing 5.3.5: The *SWI-PROLOG* implementation of the basic predicates *mode/2* and *eqmode/2*.

*DELAYS &
TIMEOUTS*

To explicitly represent time and realize timeout mechanisms, the *BFQL* provides various predicates to install and evaluate timers. For example, Listing 5.3.6 shows the *SWI-PROLOG* implementation of the predicate *timeout/2* that can be used to create timers using the dynamic predicate *timer/2* and to evaluate the time that has elapsed since their installation. The predicate fails and asserts a timer to the fact base at its first evaluation, in consecutive calls it checks if a certain time has expired since a timer's installation until it finally succeeds and retracts the timer again as soon as the timeout has expired.

```
timeout(Name, _) :-
    timer(Name, Time), !,
    now(Now), Now > Time,
    retractall(timer(Name, Time)).

timeout(Name, Delay) :-
    now(Now), Time is Now + Delay,
    assertz(timer(Name, Time)), fail.
```

Listing 5.3.6: The *SWI-PROLOG* implementation of the commonly used timeout predicate *timeout/2*.

To realize inter-process communication and information exchange between concurrent processes via the *PROLOG* fact base, the *BFQL* defines a variety of predicates that are used to produce and consume signal events. For example, Listing 5.3.7 shows the *SWI-PROLOG* implementation of the signal exchange predicates *signal/2* and *detect/2*.

SIGNAL
EXCHANGE

```

signal(Mode, Name) :-
  forall((fsr(Record),
         val(type, signal, Record),
         val(mode, Mode, Record),
         val(name, Name, Record)),
        del(Record)), now(Time),
  add([type : signal, mode : Mode, name : Name, time : Time]).

detect(Mode, Name) :-
  fsr(Record),
  val(type, signal, Record),
  val(mode, Mode, Record),
  val(name, Name, Record),
  del(Record).

```

Listing 5.3.7: The *SWI-PROLOG* implementation of the signal predicates *signal/2* and *detect/2*.

Temporal Event Relations

Quantitative and qualitative temporal relations between events are key concepts to be represented in order to combine several events from multiple modalities. Qualitative time addresses temporal relations between events and the ordering of event such as precedence, succession, and simultaneity. The *BFQL* defines predicates that are based on *interval temporal logic* (Allen, 1981, 1983, 1984; Allen and Hayes, 1990; Allen and Ferguson, 1994; Allen, 2013) which is well suited to argue about the *qualitative temporal relations* between events and actions in time. Figure 5.3.5 shows an illustration of the possible qualitative temporal relations between events. As example, Listing 5.3.8 shows the *SWI-PROLOG* implementation of the predicate *during/2* which checks if an event takes place during another one.

QUALITATIVE
TEMPORAL
RELATIONS

```

during(Event1, Event2) :-
  val(time, Time1, Event1), val(time, Time2, Event2),
  val(dist, Dist1, Event1), val(dist, Dist2, Event2),
  val(life, Life1, Event1), val(life, Life2, Event2),
  Start1 = Time1 - Dist1, Start2 = Time2 - Dist2,
  End1 = Start1 + Life1, End2 = Start2 + Life2,
  Start1 > Start2, End1 < End2.

```

Listing 5.3.8: The *SWI-PROLOG* implementation of the temporal relation predicate *during/2*.

Qualitative relations are only able to express the ordering of events, such as temporal succession, overlapping, or simultaneity. However, they are not able to express exact *quantitative temporal relations* between events, such as the temporal distance of two events or the relation

QUANTITATIVE
TEMPORAL
RELATIONS

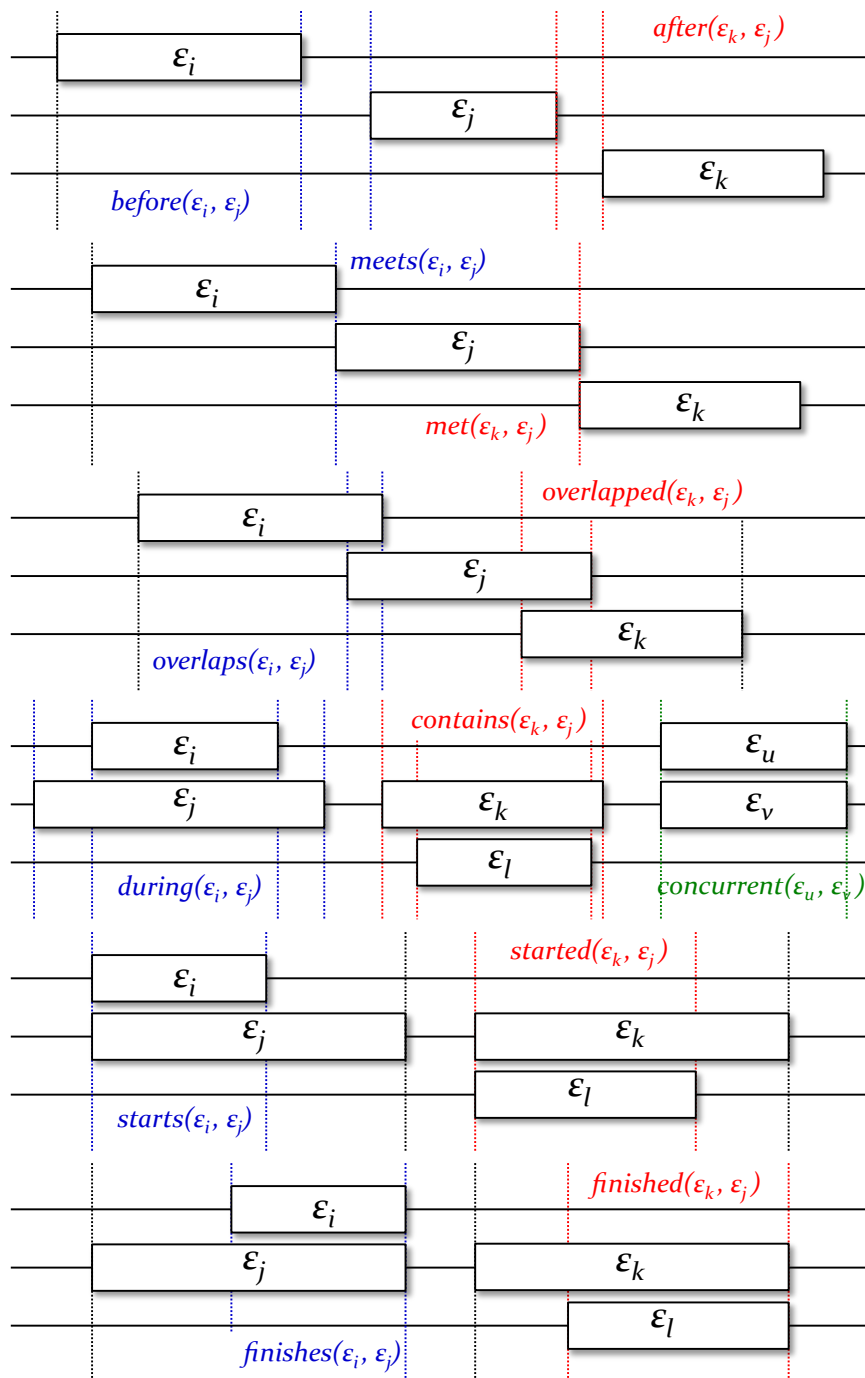


Figure 5.3.5: Some pictorial illustrations of different qualitative temporal relations between events.

of the start or end times of an event to absolute points in time. Therefore, the *BFQL* defines a number of predicates to evaluate quantitative temporal relations between events. For example, Figure 5.3.6 shows an exemplary use of the predicate *rel_dist/3* to infer the relative temporal distance of the two events, in this case the time an object was moved.

Events of the same modality, user, and device or multiple events that satisfy a certain constraint can be linearly ordered. In this case, one might be interested in the oldest, latest, or

$$\begin{array}{l}
 \text{?- } rel_dist \left(\left[\begin{array}{l} \text{TYPE: } event \\ \text{NAME: } user \\ \text{MODE: } touch \\ \text{TIME: } 8500 \\ \text{DATA: } \left[\begin{array}{l} \text{TYPE: } start \\ \text{POS: } \left[\begin{array}{l} X: 368 \\ Y: 117 \end{array} \right] \\ \text{NAME: } 1 \end{array} \right] \\ \text{LIFE: } 50 \\ \text{DIST: } 50 \\ \text{CONF: } 1.0 \end{array} \right], \left[\begin{array}{l} \text{TYPE: } event \\ \text{NAME: } user \\ \text{MODE: } touch \\ \text{TIME: } 11400 \\ \text{DATA: } \left[\begin{array}{l} \text{TYPE: } stop \\ \text{POS: } \left[\begin{array}{l} X: 139 \\ Y: 457 \end{array} \right] \\ \text{NAME: } 1 \end{array} \right] \\ \text{LIFE: } 50 \\ \text{DIST: } 50 \\ \text{CONF: } 1.0 \end{array} \right], Distance \right). \\
 \text{Distance} = 2850.
 \end{array}$$

Figure 5.3.6: The exemplary use of the quantitative temporal relation predicate *rel_dist/3* in a query.

just a random event from such a set of events. Additionally, we want to make statements about the ordering of these events, such as determining followers and ancestors or to express neighborly relations between them. The *BFQL* defines various predicates to evaluate such ordering and neighborhood relations. For example, Listing 5.3.9 shows the *SWI-PROLOG* implementation of the predicate *oldest/3* which infers the oldest event from a set of events that fulfill a particular requirement.

```

oldest_of_list(R, [R]):- !.
oldest_of_list(R, [H|_]) :-
    oldest_of_list(L, _),
    ( before(L, H), !, R = L ; before(H, L), !, R = H ).

oldest(Template, Generator, Element) :-
    bagof(Template, Generator, List),
    oldest_of_list(Element, List).

```

Listing 5.3.9: The *SWI-PROLOG* implementation of the ordering relation predicate *oldest/3*.

Generalized Quantification

The *BFQL* provides a number of higher-order predicates that implement *generalized quantifiers* (Mostowski, 1957; Lindström, 1966). Generalized quantifier theory is a formal, set-theoretic framework which is particularly suitable for the semantic representation and analysis of *quantification expressions* and *set relations* as they occur rather frequently in natural language. People, for example, naturally and frequently use such quantifier expressions for talking about the quantity of things and to describe numerical or proportion relations between sets of individuals or events, such as “a dozen of”, “less than five of”, “at least half of”, “a third of”, “the majority of” or “the largest portion of”. To cope with these kinds of quantification expressions, philosophers, logicians, and linguists developed the theory of generalized quantifiers, thus, extending the expressive power beyond the standard first-order logic quantifiers “for all” and “for some”. Generalized quantifiers, by this, universalize the notion of a

QUANTIFICATION
EXPRESSIONS

quantifier in a precise, formal, mathematically well-defined way (Barwise and Cooper, 1981; Montague, 1988; Keenan and Westerståhl, 2011). Their major advantage is the close correspondence between the syntax of *logic quantification* formulas and queries and *natural language quantification* (D’Alfonso, 2011). The generalized quantifier predicates of the *BFQL* exploit this syntactic closeness to offer an expressive, declarative, intuitive, and elegant method for the evaluation of quantification constraints between sets of individuals in the fact base and events in the event history, an otherwise often procedurally solved problem.

GENERALIZED
QUANTIFIER
PREDICATES

The set-theoretically defined generalized quantifiers mostly evaluate a binary relation between two sets of solutions. The first solution set is referred to as *restrictor* because it is generated by a goal constraining the domain of quantification. The second set is called *nuclear scope* because it expresses the set with which the restrictor is confronted in order to determine if the relation is satisfied (D’Alfonso, 2011). Their implementations often have a tripartite structure using a common notation used in logic programming which is referred to as *three-branch quantifier* (Colmerauer, 1978; Warren and Pereira, 1982). In this notation, the functor is called the *quantification*, the first argument is called the *template*, usually a term with unbound variables, the second argument is referred to as *range* and the last as *scope*. Listing 5.3.10 shows the exemplary *SWI-PROLOG* implementations of the generalized quantifiers *formost/3* and *forfraction/4* based on the solution collection predicate *collect/3* which falls back on the standard *SWI-PROLOG* meta-predicate *bagof/3*. The implementation of *forfraction/4* additionally specifies a fourth argument for the fraction. There exist many more, actually infinitely many, imaginable quantifiers (Peters and Westerståhl, 2006).

```
collect(Template, Generator, Collection) :-
    bagof(Template, Generator, List)
    *-> Collection = List ; Collection = [].

formost(Template, Generator, Condition) :-
    collect(Template, (Generator, Condition), Range),
    collect(Template, (Generator, \+(Condition)), Scope),
    length(Range, R), length(Scope, S), R > S.

forfraction(Fraction, Template, Generator, Condition) :-
    collect(Template, Generator, Range),
    collect(Template, (Generator, Condition), Scope),
    length(Range, R), length(Scope, S), R \== 0, Fraction is S/R.
```

Listing 5.3.10: The *SWI-PROLOG* implementation of the *BFQL* quantifiers *formost/3* and *forfraction/4*.

GENERALIZED
QUANTIFIER
TRANSLATION

Natural language quantifications can straightforwardly be translated into logic quantification formulas, or *PROLOG* terms, respectively, that are based on the three-branch quantifier notation (Warren and Pereira, 1982; Pereira, 1983a; Cooper *et al.*, 1993). This is especially useful to translate natural language quantification constraints that an author might consider as reasonable for multi-modal ambiguity resolution. For example, assuming that a speech event representing the user’s utterance has already been extracted from the event history and unified with the variable *v*. Then the author could informally formulate the natural lan-

guage constraint “Let ϑ be the name of a beach photo to which the user looked most of the time during v ”. This can be reformulated to a more formal natural language quantification query using the formulation “Find solutions for ϑ such that the largest portion of the user’s gaze fixations to photos of a beach during the event v are exactly to the photo with name ϑ ”. Figure 5.3.7 illustrates how this natural language quantification query can finally be translated to a syntactically rather similar looking logic quantification query using the quantifier predicate *forlargest/3* and some of the already mentioned predefined *BFQL* predicates. The color coding in the spoken sentence (Figure 5.3.7 (A)) and the logic query expression (Figure 5.3.7 (B)) shows which natural language parts are translated to the respective goals and clauses.

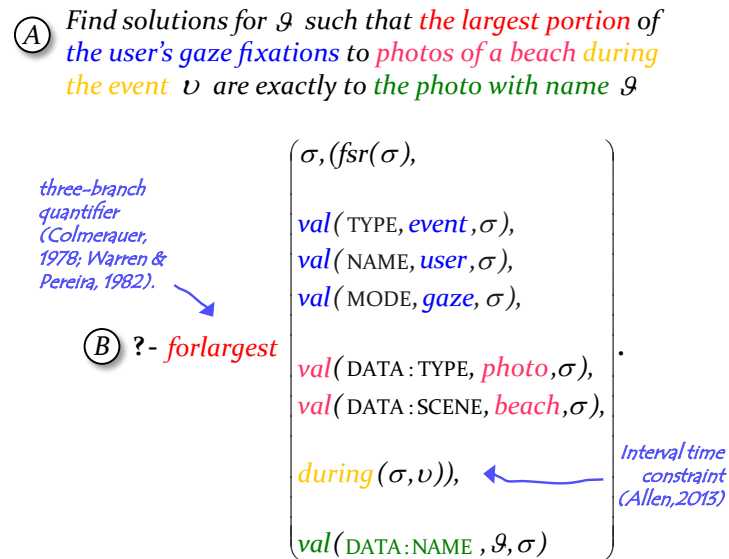


Figure 5.3.7: An example how a natural language quantification can be translated into a logic query.

The example in Figure 5.3.7 illustrates that, besides the intuitiveness of generalized quantifiers due to their syntactic closeness to natural language, the true strength of the *BFQL*’s quantifier predicates is that the unification and backtracking mechanism of the *PROLOG* inference engine can be exploited to collect solutions for uninstantiated variables in the corresponding goals of the logic quantification query. Based on the aforementioned example, this mechanism is illustrated in more detail by Figure 5.3.8 which shows two of the many ways how a logic quantification query with free variables can be used for the collection of solutions that can be used for the multi-modal disambiguation of the user’s input. We assume that the surface table in front of the user is populated by three photos, two of which show a scene with a beach and one shows a scene with a forest (Figure 5.3.8 (A)). The user is then speaking the ambiguous utterance “Tell me, where is this beach” while looking at the photos on the surface table (Figure 5.3.8 (B)). Generalized quantification is used to disambiguate this referring expression by collecting the gaze fixation events during the utterance that match the feature description in the utterance, that means their target is a photo showing a beach. For that purpose, the quantifier first collects all gaze events during the user’s utterance (Figure 5.3.8 (C)). Then it considers only those of these gaze events whose targets are photos showing scenes with beaches (Figure 5.3.8 (D)). In the first case (Figure 5.3.8 (C)), the query

GENERALIZED
QUANTIFIER
APPLICATION

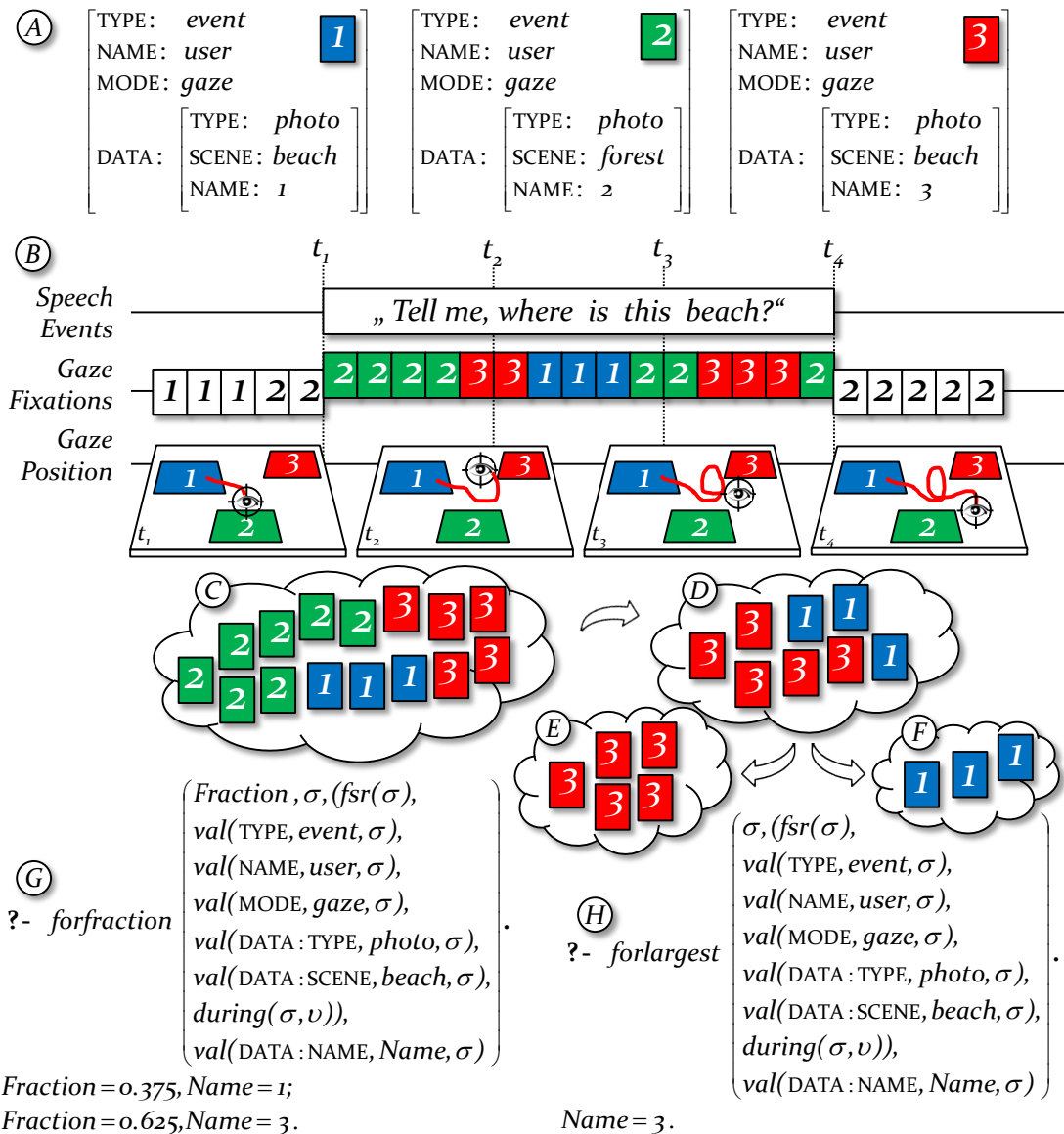


Figure 5.3.8: An exemplary usage of the generalized quantifiers *forfraction/4* and *forlargest/3*.

with the *forfraction/4* quantifier is then used to find all possible fractions of these gaze events that share the same name and instantiates the variable *Fraction* with the relative size of these sets and the variable *Name* with the name of the corresponding photos. Because there exist two different photos showing beaches that have been looked at during the utterance, the *PROLOG* engine returns two solutions for this query. First, the photo with name 1 has been the target of 37.5% of the gaze events (Figure 5.3.8 (F)) and, second, the photo with name 3 has been target of 62.5% of the gaze events (Figure 5.3.8 (E)). In the second case (Figure 5.3.8 (H)), the *forlargest/3* predicate is used to find out the name of the photo to which the largest portion of matching gaze events during the user’s utterance refers to. As the name suggests, this predicate is returning at most a single solution and the query in this case comes up with the photo with name 3 as solution (Figure 5.3.8 (E)). The example illustrates how elegant, versatile, and expressive logic generalized quantification queries can be used for ambiguity

resolution.

5.4 Coordinating Functions and Processes

As identified in Section 3.2, a key modeling task is the close coordination, which means interleaving, synchronization, and prioritization, of the various concurrent, hierarchical, incremental, and reciprocal aspects of behavior as well as the proper dialog and interaction flow management. The *BFML* meets this task with *Behavior Flow State-Charts (BFSCs)*, a specially designed, hierarchical and concurrent state-chart variant that extends the simpler, flat, and non-parallel predecessor (Gebhard *et al.*, 2003a) with methods for the hierarchical refinement and parallel decomposition of a model and an automatically maintained interaction history. Diverse transition types allow representing the temporal, conditional, probabilistic, and concurrent aspects of behavior, interaction, and dialog. States and transitions contain *BFGL* commands that execute *BFSL* activities and *BFQL* queries. They are influenced by modeling approaches based on finite-state automata (McTear, 1998; Johnston and Bangalore, 2005; Iurgel, 2006; Bourguet and Chang, 2008; Raux and Eskénazi, 2009), state transition networks (Wasserman, 1985; Latoschik, 2002, 2005), petri-nets (Navarre *et al.*, 2005; Chao and Thomaz, 2011, 2016), and other state-chart dialects (Skantze and Moubayed, 2012; Brusk *et al.*, 2007; Kronlid and Lager, 2007). *BFSCs* are an expressive and practicable visual modeling method facilitating extensibility, reusability, and clear structuring of a model.

BEHAVIOR
FLOW
STATE-CHARTS

5.4.1 States, Transitions and Variables

A *BFSC* consists of different types of *states* and *transitions* which are graphically represented by differently shaped *nodes* and *edges* in the corresponding state transition diagram. Nodes and edges are labeled with *BFGL statements* and *expressions* as well as graphical elements, such as *start node markers*, *end node markers*, or *evaluation policy markers* that determine and reflect the execution semantics of the corresponding states and transitions. A node may have an arbitrary number of incoming edges, however, the allowed number and valid combinations of outgoing edge have to obey to certain syntactic and semantic constraints.

ELEMENTS &
EXECUTION

The *BFGL* statements of a node together make up a small executable program segment, very similar to a procedure of a general purpose procedural programming language. The program segments of all nodes within a *BFSC* and the connections that are established by the different edges between these nodes together constitute a larger static program specification. The actual sequence of *BFGL* statements during a specific execution run of a *BFSC* is then determined at runtime by the evaluation of the conditions guarding the transitions. *BFSCs* are *static* executable specifications and do not contain any language constructs for the *dynamic* creation of states and transitions or memory allocation during their execution.

Definition of Nodes

The simplest type of node in a *BFSC* is a *basic node* that, for historical reasons, is also called *scene node* (Gebhard *et al.*, 2003a, 2008, 2012) because it allows playing back a *scene activity*

THE SYNTAX OF
BASIC NODES

from a *scene script* written in *BFSL*. An ordinary basic node is graphically represented by a single-lined circle (○) that is labeled with, first, a not necessarily unique *node name*, used to describe its function or task, and, second, a unique *node identifier* which enables the unambiguous identification and referencing of the node. If the node is also a *start node*, then it is labeled with a triangle (▷) and if it represents an *end node*, then it is depicted with a double-lined circle (⊙). A basic node represents a simple state in the *BFSC* and can specify a program consisting of *BFGL* statements, such as, among others, type and variable definitions or assignments, commands to play back behavioral activities specified in *BFSL* or execute queries to the *PROLOG* fact base using *BFQL* predicates as well as calls to author- or pre-defined functions, like history and configuration operators or methods of plug-in modules in the underlying implementation language.

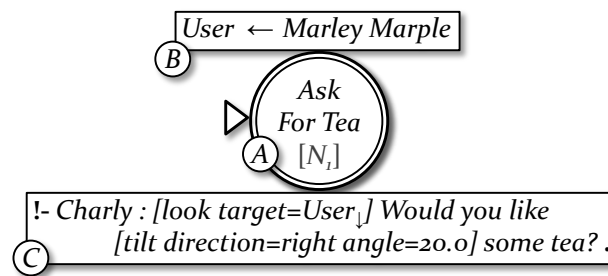


Figure 5.4.1: A basic node (Ⓐ) labeled with a variable definition (Ⓑ) and a playback command (Ⓒ).

Figure 5.4.1 shows an exemplary definition of a basic node in the conceptual notation of *BFSCs*. The node has the unique node identifier N_i , the node name *AskForTea* and is marked as both, start node (▷) and end node (⊙) (Figure 5.4.1 Ⓐ). It is labeled with a local variable definition (Figure 5.4.1 Ⓑ) and a playback command for a behavioral activity (Figure 5.4.1 Ⓒ). When executed by a state-chart process of the *IRE*, then this process first initializes the local variable *User* with the string value “*Marley Marple*” and afterwards lets the agent Charly execute the multi-modal utterance activity “[*look target=User₁*] *Would you like [tilt direction=right angle=20.0] some tea?*” in which he is looking at Marley with a questioning head pose while asking her if she would like to drink some tea.

An ordinary *super node* is graphically represented by a single-lined square (□) that can be labeled with the graphical and textual language constructs that may also be used with basic nodes. If the super node is a *start node*, then it is, just like a basic node, labeled with a triangle (▷) and if it represents an *end node*, then it is drawn as double-lined square (⊞). In addition, it extends the functionality of a basic node with the extra possibility to create a *hierarchical structure* on the *BFSC*. To achieve this, a super node may, in contrast to a basic node, contain an arbitrary number of *nested subnodes*, which can be basic nodes or super nodes themselves and that together constitute one or more nested *BFSCs*. Any subset of the nested subnodes of a super node may be declared as the *start node set* of that specific super node by marking them with start node markers (▷). They serve as starting points for the execution of the nested *BFSCs* and have to be executed in concurrent processes by the *IRE*. The hierarchy of super nodes creates a *variable scoping* such that variable definitions of a super node are visible

and accessible by all nested nodes but not its parent super nodes and their additional nested child nodes. The super node hierarchy and variable scoping mechanism creates a hierarchy of local execution contexts that can be used for the context-sensitive reaction to user input events and context changes, similar to modeling approaches using hierarchical task networks. In addition, the *hierarchical refinement* of the model helps to clearly structure a *BFSC* and thus facilitates maintainability and reusability.

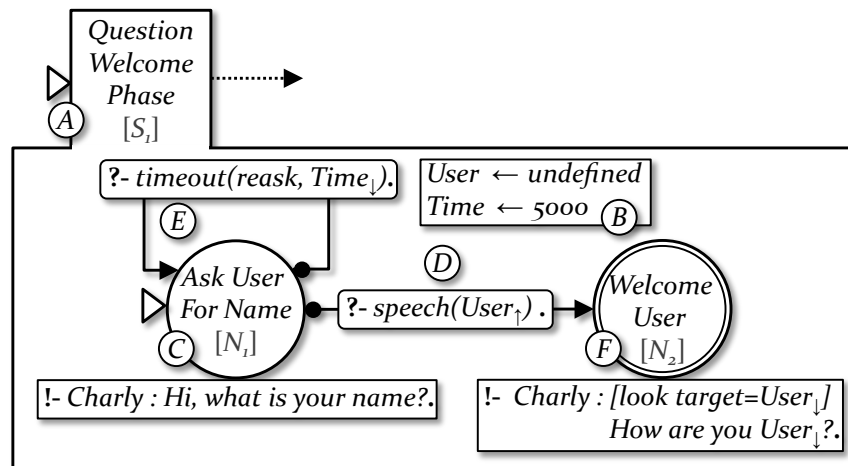


Figure 5.4.2: A super node (A) defining local variables (B) and containing a short dialog phase with a single question-answer pair between the agent and the user in a nested subnode of the *BFSC* (C-F).

Figure 5.4.2 shows an example of a super node in the conceptual notation of *BFSCs*. The super node with identifier S_1 and name *QuestionWelcomePhase* (Figure 5.4.2 A) encapsulates a nested *BFSC* that implements a very simple question and welcome phase between the agent and the user. The super node defines two local variables (Figure 5.4.2 B), first the initially undefined variable *User* to represent the user's name and, second, a variable *Time* to represent a specific timeout value of 5000 milliseconds. The short dialog starts with the question for the user's name (Figure 5.4.2 C) by playing the respective utterance activity in the start node with identifier N_1 and name *AskUserForName*. The execution of this node is repeated every 5000 milliseconds triggered by a timeout guarded transition (Figure 5.4.2 E) until the user answers with an utterance that provides his or her name. The verbal user input is detected via a query to the *PROLOG* fact base using the *BFQL* predicate *speech/1* (Figure 5.4.2 D). Then the agent uses the information about the user's name to greet the user by his or her name in the end node N_2 (Figure 5.4.2 F) before the execution continues with the edges at S_1 .

Definition of Edges

An *edge* connects two nodes in a *BFSC* and specifies the conditions under which the transition between the corresponding states may be taken. This transition is then concatenating their individual executable *BFG*L specifications to a single program. Besides some differences, there exist a number of common syntactical and semantical characteristics of all types of edges. An edge is always a directed and optionally labeled arrow (\Rightarrow) that is connecting

exactly two nodes in a *BFSC* which are called the *source node* and the *target node* of this edge. An edge that connects a node with itself is called *self-loop edge*. An edge may be labeled with a *transition guard* that determines the preconditions that must be fulfilled for taking the transition. A guard may be, for example, a temporal, conditional, or probabilistic constraint that needs to be satisfied to enable the corresponding transition. In addition, an edge can be labeled with an *transition marker* that specifies the evaluation policy of the transition guard. It determines *when* and *how* the guard comes into effect, that means *how often* and *by which* process of the *IRE* the guard will be evaluated during the execution of a *BFSC*. The semantics of different types of transition guards and evaluation markers can finally be combined in order to realize a variety of branching strategies and interruption policies within the *BFSC*.

THE TYPES OF
TRANSITIONS &
THEIR GUARDS

Figures 5.4.3 to 5.4.7 show examples of edges with different types of guards that determine when the respective transitions may be taken, for example, immediately and unconditionally, with a certain probability, after a specific timeout period has elapsed, or if a particular conditional expression is fulfilled.

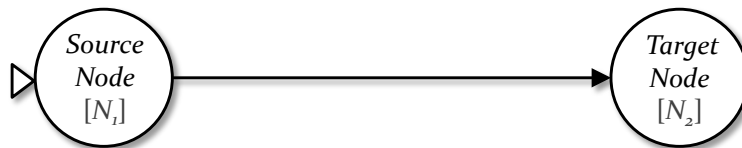


Figure 5.4.3: Two nodes connected with an *epsilon edge* that defines an unconditional transition.

Figure 5.4.3 shows an unlabeled edge between two nodes which is referred to as *epsilon edge*. Such an edge represents an immediate and unconditional transition from its source node to its target node. It can directly be taken once the process executing the source node is scheduled the first time after finishing the *BFGL* program of the source node. If a source node has multiple outgoing epsilon edges or any additional enabled edges of another type, then one of these transitions is chosen *nondeterministically*. Epsilon edges can be used by an author to create a sequential structure of the behavior and interaction flow and to specify the order in which computation steps, behavioral activities, and logic queries are executed. They are useful to make the model clearly arranged and to facilitate the manageability and readability of the model.

Figure 5.4.4 shows an edge that is guarded by a *timeout guard* and which is referred to as *timeout edge*. It is labeled with a *timeout value* and a *measuring unit*, such as, for example, *ms* for milliseconds (Figure 5.4.4 (A)) or *s* for seconds (Figure 5.4.4 (B)). It represents a scheduled transition that is taken with the desired delay when the executing process is scheduled after the corresponding time interval has expired since the guard has been evaluated for the first time after the *BFGL* program of the source node has been finished. If several timeout edges or additional epsilon or condition edges can be taken at the same time, then the *IRE* must select one of these transitions *nondeterministically*. Timeout edges, explicitly representing time in the model, are used to control and regulate the timing and temporal flow of a behavior and interaction model's execution, in particular, to schedule the playback of behavioral activities,

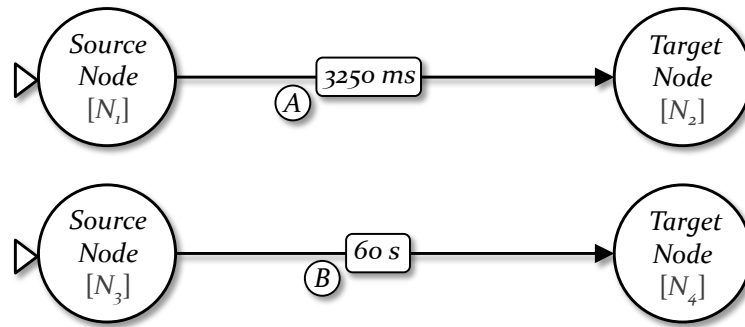


Figure 5.4.4: Two nodes connected with a *timeout edge* that defines a scheduled transition.

the execution of logic queries, and other computation steps.

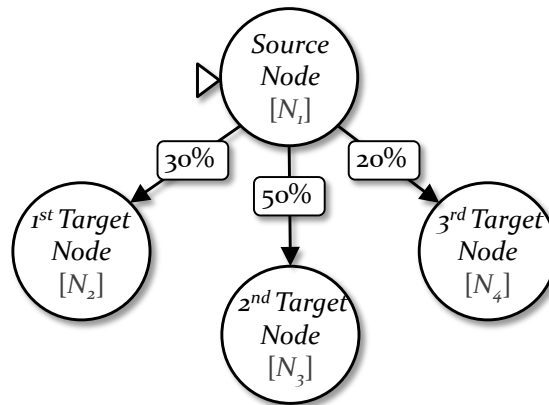


Figure 5.4.5: Nodes connected with a set of *probability edges* that define probabilistic transitions.

Figure 5.4.5 shows a set of edges that are guarded by *probability guards* and which are referred to as *probability edges*. They are labeled with values between zero and one hundred followed by a percentage sign (%) and represent transitions that are taken with the respective probability. A node must either have solely outgoing probability edges or none and the sum of the probabilities of all probability edges of a node have to sum up to 100%. Once the execution of the source node's *BFGL* program has finished, one of the outgoing transitions is chosen and taken according to their specified probabilities. Probabilistic edges are used to create some degree of randomness and desired nondeterminism during the execution. This facilitates to produce some unpredictability and variability in the behavior or dialog of a social agent which helps to make the agent appear more vivid, natural, and credible.

Figure 5.4.6 and Figure 5.4.7 show different edges that are referred to as *condition edges*. They represent conditional transitions that can, for example, be guarded by logical and arithmetic (Figure 5.4.6 (A)) or comparison expressions (Figure 5.4.6 (B)), function calls to built-in or user-defined methods in the underlying implementation language (Figure 5.4.6 (C)), or logic queries to the *PROLOG* fact base using *BFQL* predicates (Figure 5.4.7 (A), Figure 5.4.7 (B)). Condition edges are used to define the branching structure in the *BFSC* which describes the

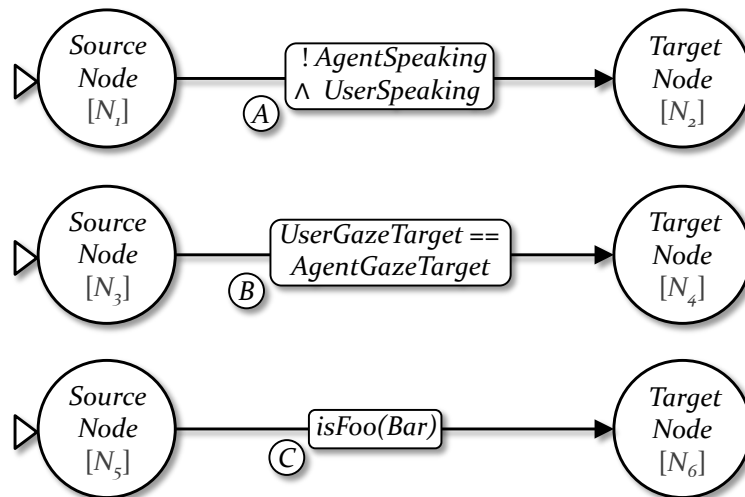


Figure 5.4.6: Nodes connected with different *condition edges* that define conditional transitions.

different possible ways the behavior and interaction flow can take in reaction to changes of environmental conditions, external and internal events, or user interactions as well as conclusions reasoned from context knowledge or data retrieved from external modules.

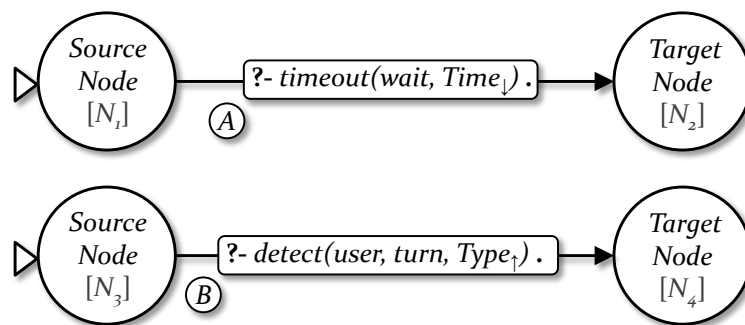


Figure 5.4.7: Nodes connected with different *condition edges* that are labeled with logic queries.

THE TYPES OF
MARKERS &
THEIR USAGE

The evaluation of a transition guard determines if the respective transition may be taken or not at the moment of this evaluation. An *evaluation policy* of a transition is used to control *when, how often* and *by which process* the transition guard may be evaluated at all. This effectively determines when the respective transition can really be taken and if other processes can be interrupted and terminated preemptively in order to realize this transition. The evaluation policy of a transition is specified with a *transition marker* which can be either a filled circle (●) or a filled star (★) at the source node of the corresponding edge. Figure 5.4.8 shows a simple exemplary *BFSC* that illustrates the use of the different transition markers to define the corresponding evaluation policies.

Edges that are neither marked with a star nor a circle are referred to as *transient transitions*. The conditions of these their guards are checked only once and only by the process that actually executes the source node when this process is scheduled and only after the *BFG*

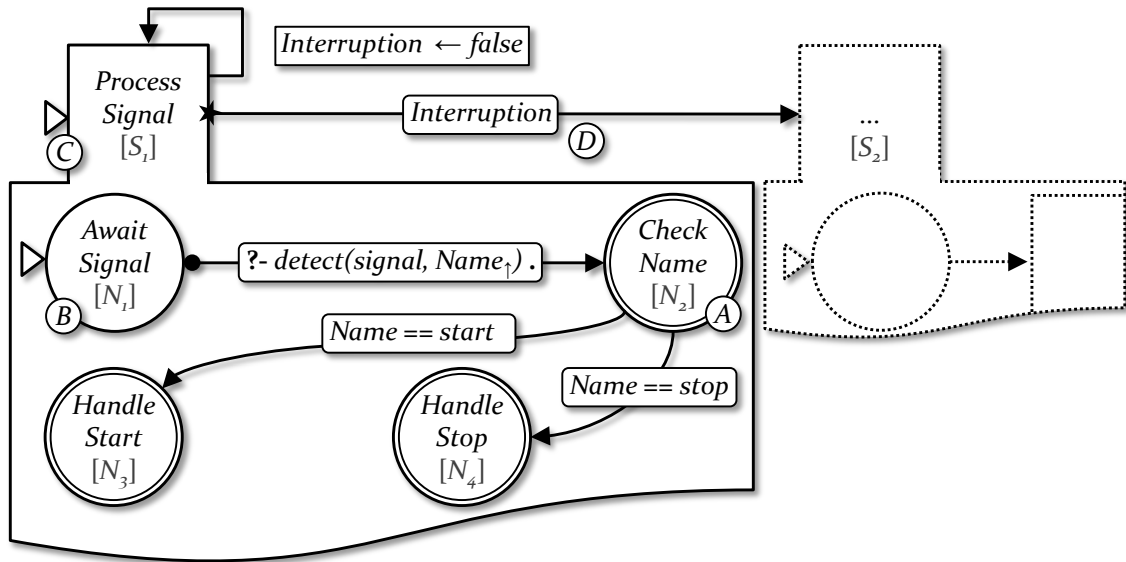


Figure 5.4.8: An exemplary *BFSC* with transitions that have different evaluation policies.

program of the source node has been fully executed. If a transient transition is not satisfied after the execution of the source node, then the executing process terminates immediately in this step. Therefore, a node that exclusively has guarded transient transitions must be marked as end node using the double lined circle or square syntax (\odot, \square). The *BFSC* depicted in Figure 5.4.8 shows examples of such transient transitions from node N_2 to the nodes N_3 and N_4 (Figure 5.4.8 (A)).

Edges that are marked with a filled circle (\bullet) denote *persistent transitions*. These transitions are repetitively but *lazily* checked after the program of the source mode has been finished. They are exclusively checked by the process that executes the source node whenever this process is scheduled and are then taken in the same step if they are satisfied. Since it is not deterministic when the process that executes the source node is scheduled by the *IRE* it cannot be guaranteed that the transition is taken whenever it is satisfied because it could be missed due to late scheduling. The *BFSC* depicted in Figure 5.4.8 shows an example of such a persistent transition from node N_1 to node N_2 (Figure 5.4.8 (B)).

Edges that are marked with a filled star (\star) denote *interruptive transitions*. These transitions may be *eagerly* evaluated by any process of the *IRE* whenever such a process executes a *step*. Such a step may be any computation or operation that can modify the *IRE*'s information and configuration state, such as executed states, variable values, or the history content. It represents a critical section that has to be accessed mutually exclusive through the interleaving of the *IRE*'s processes in order to guarantee the consistency of the execution state. According to this *step interleaving semantics* of *BFSCs*, an interruptive transition is evaluated by each process that has just entered a state, recently executed a *BFG*L statement, or evaluated a transition guard, and even when it is scheduled after it has already finished its current node's *BFG*L program. If an interruptive transition is satisfied, then the transition is immediately

taken in the same step while all effects of this transition are then soonest observed in the next step to avoid infinite chain-reactions with this execution semantics. In contrast to persistent transitions, however, the change of the guard of an interruptive transition can never be missed because it is evaluated in each step, such that the edge is always taken if it is satisfied. The *BFSC* depicted in Figure 5.4.8 show an example of such an interruptive transition from super node S_1 to node S_2 (Figure 5.4.8 ©). The execution of the nested *BFSC* of S_1 is immediately terminated and the execution proceeds with the target node S_2 in the next step when the guarding variable expression *Interruption* has become *true* (Figure 5.4.8 ⓓ).

Usage of Variables

USAGES OF A
VARIABLE NAME

The *BFGL* allows the definition of state-chart variables that are mainly used for representing values and referring to these values when using the name of the variable in statements and expressions. However, an additional key role of variables in our modeling framework is the exchange of values between the logic *PROLOG* fact base, the environment of the *BFSC*'s *IRE*, and *BFSL* placeholders in behavioral activity templates. To meet this requirement, *BFSC* variables can be used with two different meanings, depending whether they occur in an ordinary *BFGL* expression or statement, or are used as part of a logic predicate in a *BFQL* query or the specification of a behavioral activity in the *BFSL*. On the one hand, the variable can be used *by value*, that means, with the result that each occurrence of the variable name within a *BFQL* query or *BFSL* specification is replaced by the *IRE* with the current value of the variable during the execution of the respective *BFGL* statement or expression. This is the case if an author wants to hand over the value of the variable as an argument to a behavioral activity or logic query. On the other hand, a variable can also be used *by reference*, that means as a reference to the name of the variable, for example, when the author has the aim to infer a substitution for the variable such that a specific *BFQL* query is satisfiable.

NOTATIONS OF
VARIABLES

The aforementioned possibilities or meanings of a variable usage manifest themselves in different notations for variables in the conceptual syntax of *BFSC*s. When referring to the value of a variable X in a *BFQL* query or a *BFSL* specification, which is basically between any occurrence of an opening $?-$, $!-$, $!=$ or $!~$ and a closing $.$, then the variable name X is annotated with a *downward directed arrow* index (X_{\downarrow}). In contrast, if an author explicitly wants to refer to the name of the variable in a *BFQL* query, with the intention that the *PROLOG* engine instantiates the variable with a *substitution*, then this is indicated with an *upward directed arrow* index (X_{\uparrow}). In this case, the variable X is considered by the *PROLOG* engine as an ordinary unbound *PROLOG* variable which can be instantiated once for each solution of the *BFQL* query. If the *PROLOG* engine can infer one or more substitutions for the variable X , such that the *BFQL* query is satisfied, then one of these substitutions is *nondeterministically* chosen by the *IRE* and used to replace the current value of the *BFSC* variable X with the new inferred value. The inference of a substitution for the *PROLOG* variable X thus either results in a re-instantiation of the *BFSC* variable X or is simply ignored if no such *BFSC* variable is defined. If the variable X is used in any other *BFGL* expression or statement, such as, for example, a variable definition or assignment, in a logical or arithmetic expression, or as argument of a

function call, then the variable can simply be denoted with its identifier without any arrow index (X). In these cases, the variable is used as in other procedural or imperative programming languages, that means that it is replaced by its value during evaluation if it is used as a variable expression and its value is updated if used on the left hand side of an assignment.

5.4.2 Decomposition and Synchronization

BFSCs apply the modeling principles of *modularity* and *compositionality* in the sense that they provide syntactical constructs that allow their *hierarchical refinement* and *parallel decomposition*. The concept of *modularity* is typically defined as a continuum describing the degree to which a system's components, in our case individual parts of a social agent's behavior and interaction model, may be *separated and recombined*. It refers to both, the tightness of coupling between components, and the degree to which the system architecture enables or prohibits the mixing and matching of components. In the context of this thesis, the term modularity mainly refers to the *modularity of mind* which means the composition of the human mind, in particular its interaction behavior, into different modules and processes which have distinct established, evolutionarily developed, and socially grounded functions (Fodor, 1983). The principle of *compositionality* is the principle that the behavior of a complex system, in our case the social agent's interactive behavior, is determined by the behaviors of its constituent parts, such as behavioral aspects, and the rules used to combine the behavior of the individual parts to the overall system, that means the agent's behavior (Pelletier, 2001).

MODULAR &
COMPOSITIONAL
STRUCTURING

In the context of modeling the interactive behavior of social agents, modular and compositional structuring of the model means that an author may separate the task of modeling the overall behavior of an agent into separate tasks of modeling individual behavioral aspects of behavior, such as, for example, different behavioral modalities, functions, processes, and levels. This is achieved by implementing the different behavioral aspects in individual *parallel BFSCs* that are then executed in *concurrent processes* by the IRE. As already mentioned in Section 3.2.2, examples for such parallel BFSCs are control processes for input event detection and preprocessing, multi-modal fusion and behavior pattern recognition, decision making regarding the interaction management and participant role assignment, or processes that manage the different aspects of the agent's expressive behavior, such as reactive role-dependent behaviors or the deliberate dialog management. These concurrent processes can be coordinated using suitable *BFGL* constructs and *BFQL* mechanisms for *inter-process communication* and *synchronization*. The proper interleaving and interplay of their computation steps then compositionally produces a plausible overall behavior of a social agent.

CONCURRENT &
SYNCHRONIZED
PROCESSES

As a consequence of the hierarchical and parallel decomposition of the model, individual behavioral aspects can be modeled and modified mainly in isolation without knowing each and every details about the implementation of the other aspects, but, just knowing the *synchronization rules* and signals that are exchanged between the respective processes. In addition, particular behavioral aspects or patterns that have already been modeled in the past can easily be reused and adapted. They can be adopted and integrated into new models or at different locations of the same model, thus, extending already existing models with new aspects.

MAINTAINABILITY
& REUSABILITY
OF THE MODELS

For example, already modeled *BFSCs* that are controlling the communication with external software modules, input devices, or output interfaces can be added to an existing *BFSC*, just like plug-in modules that are managed in a parallel process. This modular and compositional modeling approach effectively reduces the complexity of the behavior model and, thus, overcomes the respective modeling restrictions and insufficiencies of related modeling languages described in Chapter 4 that rely on only a single thread of execution in a sequential model.

Parallel Decomposition

LARGE STEP INTERLEAVING SEMANTICS *BFSCs* offer two different syntactical instruments for the purpose of *parallel decomposition*. Parallel processes are created at a node that is marked with a *start node marker* (\triangleright) or a forking construct that is specified with a *forking marker* (\blacksquare) and are regularly terminated at an end node that is marked with an *end node marker* (\odot , \square). Both instruments can be used to split the thread of execution into several separate processes that are concurrently and interleaved executed by the *IRE* according to the already mentioned *step interleaving semantics*. This semantics prescribes that individual parallel *BFSCs* are executed by concurrent processes of which only one is selected and may execute a step at any time. A step executes a critical section, that means, any computation or operation that can modify the *BFSC's* information and configuration state. The interleaving semantics guarantee the mutual exclusive access to these sections and, thus, the consistency of the execution state. It also prescribes that all processes, that are executing parent super nodes, must wait and may not execute any step until all their child processes have regularly terminated before they proceed.

DEFINING MULTIPLE START NODES The first method is the specification of *multiple start nodes* which is very similar to the definition of *orthogonal components* of more classical state-charts (Harel and Politi, 1998). By selecting multiple start nodes for a super node, an author implicitly defines several parallel *BFSCs*, each of which represents a connected component consisting of exactly those nodes that are reachable from the respective start node, eventually sharing parts with other components. During the execution of the *BFSC*, each individual *start node marker* (\triangleright) creates a new process that executes the corresponding start node and the subsequent nested *BFSC*.

Figure 5.4.9 shows an example of a super node with node identifier S_1 (Figure 5.4.9 (A)) that has three start nodes with the node identifiers N_1 , N_2 and N_3 (Figure 5.4.9 (B), (C), (D)). The process which is responsible for the execution of the super node S_1 first executes all program statements of S_1 before it creates a new child process for each start node of S_1 during a single step. These child processes then concurrently execute the three nested *BFSCs* that are defined by the reachable nodes of the respective start nodes N_1 , N_2 and N_3 . In the meanwhile, the parent process executing S_1 waits until all its child processes have terminated and afterwards continues by evaluating outgoing transitions of S_1 and proceeding with their target nodes.

THE USAGE OF FORKING EDGE CONSTRUCTS The second method to create multiple concurrent processes in a *BFSC* is the use of *forking edges*, which are very similar to the *fork constructs* of classical state-charts (Harel and Politi, 1998). They allow modelling parallel *BFSCs* on the same level of the node hierarchy without the need of changing the level of the node hierarchy which is necessary when using super

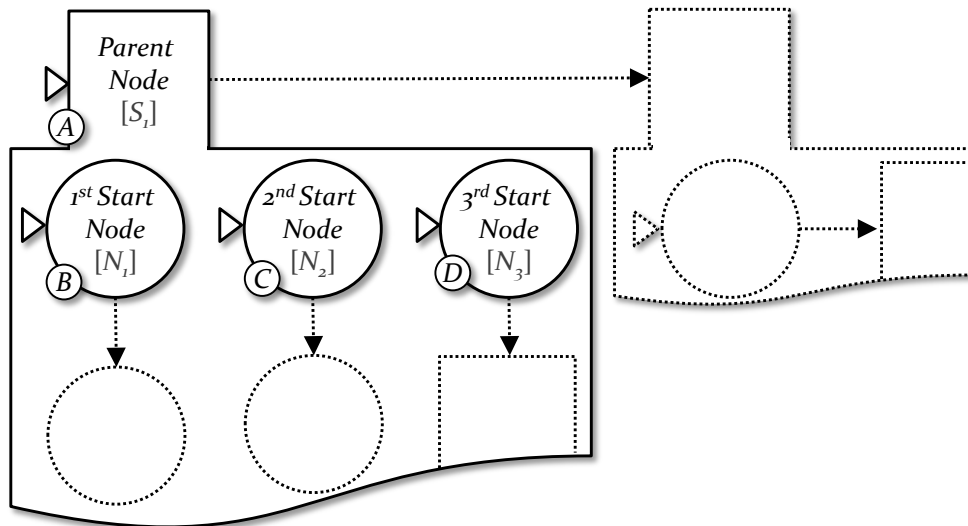


Figure 5.4.9: A super node containing several parallel nested *BFSCs* with their start nodes.

nodes with multiple start nodes for parallel decomposition. A forking edge construct consists of a single, unlabeled, outgoing edge that originates the source node and that is split, using a *forking marker* (■), into multiple edges that are ending in distinct target nodes.

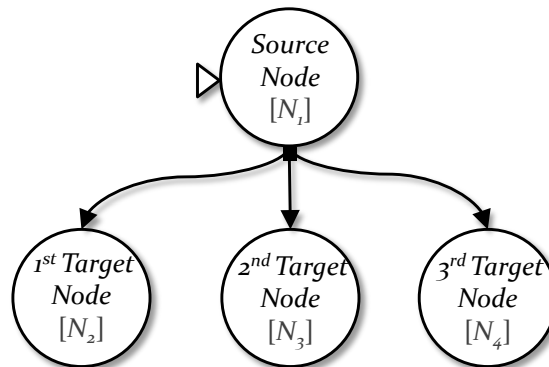


Figure 5.4.10: Several nodes connected with a forking construct that is splitting the execution.

Figure 5.4.10 shows an example of a forking edge construct that connects a source node with node identifier N_1 to three target nodes with the node identifiers N_2 , N_3 and N_4 . After the process that executes the source node N_1 has finished all program statements of N_1 it is terminated by the *IRE* and, in the same step, replaced with three new concurrent processes each of which continues with the execution of one of the target nodes N_2 , N_3 and N_4 .

Process Synchronization

The parallel decomposition of the model allows modeling individual behavioral aspects, functions and modalities in mainly insulated parallel *BFSCs*. The overall behavior of the model is consequently no longer determined by a single process only executing a single flat and serial *BFSC*, but, on the contrary, by the interleaving of multiple processes executed in parallel.

*SYNCHRONIZING
CONCURRENT
PROCESSES*

However, the individual behavioral aspects that contribute to the behavior of a social agent are rarely completely independent, but have to be coordinated and synchronized with each other. Therefore, when modeling individual behavioral functions and modalities in separate parallel *BFSCs*, the processes that concurrently execute these *BFSCs* have to be explicitly synchronized by the author in order to coordinate the respective behavioral aspects. The implemented synchronization rules determine the possible interleaving of the *IRE*'s individual processes' computation steps. *BFSCs* comprise a number of language constructs that enable an author to model different blocking or non-blocking mechanisms for the synchronization these concurrent processes (Lampert, 1986a).

THE SHARED MEMORY MECHANISM *BFSCs* allow the synchronization via a *shared memory mechanism* using the jointly accessible variables that are defined in the scope of some common super node. The step interleaving semantics of *BFSCs* prescribes a mutually exclusive access to those variables (Lampert, 1986b) in order to avoid inconsistencies. That means, any access to the variables of a super node represents a *critical section*, such that a variable can only be read or written by a single process of the *BFSC* at any point in time. The synchronization of several concurrent processes can then be achieved if one of these processes writes the value of a variable while the other processes are waiting, that means blocking, for read access. A reading process might then constantly check the value of the variable and continue with their work depending on this value.

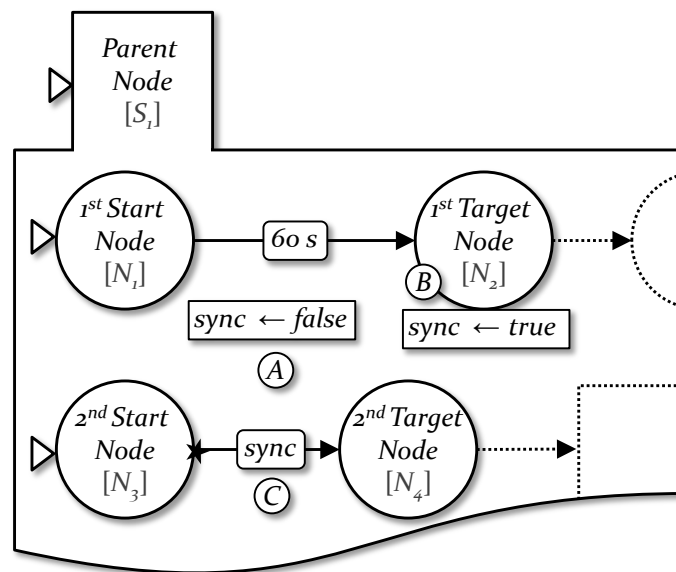


Figure 5.4.11: The usage of a shared variable named *sync* for the synchronization of *BFSCs*.

Figure 5.4.11 shows an example of the synchronization using the shared memory mechanism. The two start nodes N_1 and N_3 are simultaneously started in two parallel child processes of the process executing the super node S_1 that defines a shared variable named *sync* (Figure 5.4.11 A). After 60 seconds, the first process takes the transition to the node N_2 where it executes a *BFGL* statement which assigns the value *true* to the variable *sync* (Figure 5.4.11 B). At the same time, the second process waits in node N_3 until it can take the interruptive transition

to node N_4 , which is guarded by exactly this variable as conditional expression (Figure 5.4.11 ©). The second process then takes this interruptive transition in the next step, as soon as it is scheduled again, after the variable has been set to *true* by the first process and immediately continues with the execution of node N_4 .

In addition to the variable-based synchronization via the shared memory, the *BFGL* also comprises a built-in *state query condition* which represents a rather intuitive, state-based mechanism for the synchronization of two process. The condition allows finding out whether a certain node is currently executed, at any point in time, by directly inspecting the internal configuration state of the *IRE*. The query to the *IRE*'s configuration succeeds if the node identifier, that is handed over as argument, or any of its child nodes is currently executed by some process and fails otherwise. The synchronization of two concurrent processes can then be achieved if one of the processes enters a new state and the other process reacts to this configuration state change by taking a transition guarded by the corresponding state condition.

THE USAGE OF
STATE QUERY
CONDITIONS

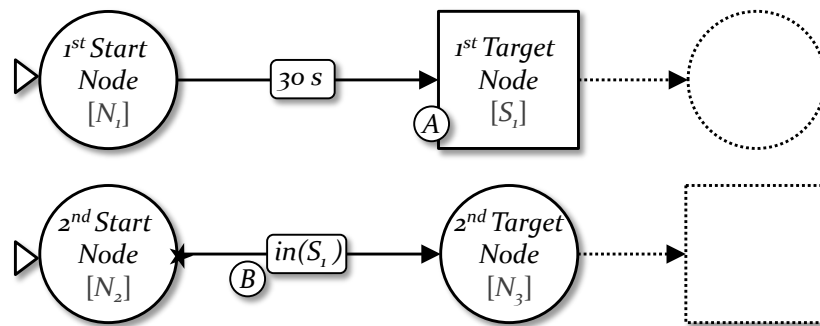


Figure 5.4.12: The usage of the built-in state condition *in/1* for the synchronization of *BFSCs*.

Figure 5.4.12 shows an example of the synchronization using the state query condition *in/1*. The two start nodes N_1 and N_2 are started simultaneously in two parallel processes. After 30 seconds, the first process takes the transition to the super node S_1 in which it remains until all child processes of S_1 have terminated (Figure 5.4.12 A). At the same time, the second process waits in node N_2 until the state query condition *in*(S_1) becomes true (Figure 5.4.12 B), which happens in the same step in which the first process enters the super node S_1 . The second process then takes the interruptive transition guarded by the state query condition and continues with the execution of node N_3 in the next step.

Finally, *BFSCs* allow the asynchronous communication and the exchange of data between two processes (Lampert, 1978) using the *fact base*. Therefore, the *BFQL* provides a set of predefined predicates that are used for signaling, sensing, and consuming special signal feature structures via the logic fact base between two particular processes or sets of processes. Figure 5.4.13 illustrates the inter-process communication using the logic fact base. The two start nodes N_1 and N_3 are started simultaneously in two parallel processes. After 3500 milliseconds, the first process takes the transition to the node N_2 where it executes the query with the *BFQL* predicate *signal/2* (Figure 5.4.13 A) which asserts an event feature structure named *sync* to the fact base (Figure 5.4.13 B). At the same time, the second process waits in

THE USAGE OF
THE FACT BASE
MECHANISM

node N_3 until it can consume exactly this signal using the query with the *BFQL* predicate *detect/2* (Figure 5.4.13 ©). The second process then takes the transition guarded by this query while extracting the feature structure from the fact base as soon as it detects the event and afterwards continues with the execution of node N_4 .

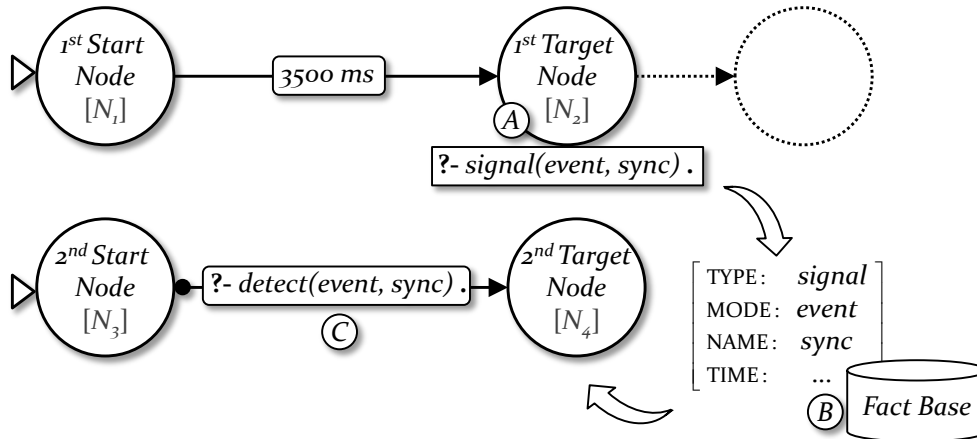


Figure 5.4.13: A signal exchange via the logic fact base using the predicates *signal/2* and *detect/2*.

5.4.3 Interruption and History Mechanism

As explained in Section 3.2.3, unforeseen and distracting context events and sudden user interactions or changing behavioral priorities or goals of the agent can rise at any time during an interaction. Some of them must be processed and reacted to as fast as possible in order to meet the real-time requirements for an immediate behavioral response. This might be necessary to give the user the impression of presence and impact, for example, when promptly stopping an utterance at an interruption attempt or following a gaze behavior at an attention shift. However, there can also be events that may be processed at some later point in time, allowing currently executed dialog phases or activities to be regularly terminated before reacting to the event, for example, when the agent is not willing to release the floor. *BFSCs* provide two adequate language constructs that can be used to realize, first, the immediate interruption and termination of behaviors or dialogs and, second, the coherent resumption of behaviors, as described in Section 3.2.3. The *interrupt mechanism* of *BFSCs* is realized with a reasonable combination of the *hierarchical refinement* and the aforementioned *interruptive transitions*. The history mechanism finds its syntactic realization in a special *history node* and a variety of built-in *history operators* that provide access to the *history memory* which is part of the *IRE's* configuration state.

Interruption Policy

INTERRUPTIVE TRANSITION POLICY As mentioned before in Section 5.4.1, a transition can have a non-interruptive, that means, either transient or persistent evaluation policy, or an interruptive evaluation policy. This evaluation policy, which can also be regarded as *interruption policy*, then determines if the source node's program has to be fully executed before the transition can be taken or if it may

be aborted in between the individual statements of the program. Consequently, interruptive transitions can be used to realize fast suspensions of a node's execution in reaction to user inputs or context events as they are necessary for immediate behavioral responses and the realization of changes of behavioral priorities and goals, as motivated in Section 3.2.3. Whenever an interruptive transition of a node can be taken, then the *IRE* may not execute any more steps, which means command statement of this node but has to take the transition as soon as possible in order to achieve an immediate reaction. If possible, even currently executed activities, such as the playback of behavioral activities or the execution of methods of the underlying programming language, should be aborted and returned from as quickly as possible. As soon as the currently executed activity has finished, the execution of the respective process directly continues with the target node of the interruptive transition.

A process that is executing a super node usually has to wait until all its child processes which are executing nested subnodes of this super node have fully terminated. Consequently, if the source node of an interruptive edge is a super node, then an immediate transition can only be realized if also all child processes of the process executing this super node are recursively terminated as fast as possible, which means, before any of these child processes can execute a statement or take a transition. To achieve this *recursive termination* of all child processes, a termination signal is propagated such that the *IRE's* processes may not take any edges or execute any more command statement of the corresponding nested nodes. This recursive termination policy implicitly introduces a *priority* on interruptive edges at different levels in a super node hierarchy. Interruptive edges at super nodes that are closer to the root in the node hierarchy of the *BFSC* have priority over any edges farther from the root. In contrast, a non-interruptive edge on a super node, has no influence on the execution of the descendant subnodes and can never cause the abortion of a child process. Consequently, such an edge can never be taken until the execution of all descendant nodes of this super node has properly terminated and all other conditions to take the edge are satisfied. This non-interruptive evaluation policy is, thus, automatically giving higher priority to any non-interruptive transition at nodes that are farther from the root.

RECURSIVE
PROPAGATION
& PRIORITIES

History Mechanism

BFSCs comprise an exhaustive, automatically maintained, and easily operated *history mechanism* comparable to the history concept of classical state-charts (Harel, 1987; Harel and Politi, 1998). During the execution of a *BFSC*, the *IRE* automatically maintains a *history memory* in which it records *runtime information* about individual execution steps of the model, such as the runtime of recently executed nodes, the last values of local variables, and a list of already executed commands. It additionally records the recently executed nested subnodes of all super nodes at the time point of the super node's regular termination or preemptive abortion due to an interruption. When modeling resumption strategies or recapitulation phases for interrupted dialog and interaction phases, then the author can fall back on this automatically gathered runtime information about the past states of the *BFSC's* execution instead of requiring the user to collect and maintain this information in a manually managed history

AUTOMATICALLY
COLLECTED
INFORMATION

node and a history condition. The super node S_1 models a simple dialog that consists of a series of different phases that are modeled by the nested super nodes S_2 , S_3 , S_4 , and so on. This simple sequential structuring is a common way to hierarchically divide a dialog or task into individual subdialogs or subtasks. If the execution of the super node S_1 is interrupted at some point in time (Figure 5.4.14 (A)) and resumed afterwards (Figure 5.4.14 (B)), for example, due to a topic switch or a distracting event, then its history node H_1 serves as starting point of a repeated execution of S_1 at a later point in time (Figure 5.4.14 (C)). The built-in history condition *HistoryContains/2* is evaluated in order to find out if the super node S_1 has been interrupted during the execution of child node S_2 , S_3 or S_4 (Figure 5.4.14 (D)). Because already finished dialog phases must not be repeated again, the implemented reopening strategy first infers the subdialog in which S_1 was interrupted and then introduces a short recapitulation dialog in R_2 , R_3 or R_4 before restarting the previously interrupted subdialog again. If, for example, the interruption took place while executing S_3 , then S_1 is resumed by, first, reopening the dialog in R_3 (Figure 5.4.14 (E)) and then restarting the subdialog in S_3 (Figure 5.4.14 (F)).

The example in Figure 5.4.14 shows, that the main advantage of the history mechanism is, on the one hand, that the automatic maintenance of the history memory releases the author of the manual collection of relevant runtime data which efficiently helps to reduce the modeling effort and time while increasing the clarity and reusability of the model. On the other hand, the interaction history provides the author with rich information about previous interactions and states of execution which significantly facilitates modeling the resumption of behavioral processes, especially the realization of reopening strategies for behaviors and recapitulation phases of dialogs. These reopening strategies are then not only chosen at random but fall back on the information in the interaction history. This makes the overall interactive performance more lively and erratic and therefore more appealing and compelling for the user. After all, the prioritization and resumption of behaviors, as described in Section 3.2.3, can be realized via, first, the immediate abortion of a behavior modeled in a super node using an interruptive transition, and, second, the use of the history mechanisms to coherently resume the behavior in this super node when it is re-executed at a later point during the interaction.

INTERRUPTION
& COHERENT
RESUMPTION

5.5 Summary and Conclusion

In this chapter, I proposed a novel modeling framework, called *BFML*, to modeling the interactive behavior of artificially and socially intelligent agents. Initially, in Section 5.1, I identified a number of important guidelines and conditions for the design of *BFML*. The framework architecture was, for example, chosen to allow an iterative and distributed development while the ensemble's modeling languages should be intuitive to use and syntactically close to natural language. Afterwards, in Sections 5.2 to 5.4, I described the theoretical foundations and definitions of the individual modeling languages. First, in Section 5.2, I showed how the template-based, textual, description language *BFSL* can be used to specify expressive and versatile multi-modal behaviors as well as credible and informed dialog content. Then, in Section 5.3, I explained how the declarative, domain-specific, *PROLOG* calculus, called *BFQL*, is used for multi-modal fusion and knowledge reasoning tasks or inter-process communi-

cation. Finally, in Section 5.4, I pointed out how *BFSCs* and *BFGL* are used for the close coordination, which means interleaving, synchronization, and prioritization, of concurrent, hierarchical, incremental, and reciprocal behavioral functions and processes.

BFML is the first such modeling approach to combine the benefits of a specially designed hierarchical and concurrent state-chart variant, a domain-specific logic calculus and a template-based behavior description language for this purpose. With respect to expressiveness, it goes beyond the state-of-the-art efforts because it successfully masters the complex coordination and interplay of the many functions and aspects that underlie interpersonal coordination and grounding. In this, it has a remarkable practicability since it uses mostly declarative and visual modeling and description languages as well as uniform representation formats. It is usable for computer experts as well as people without programming skills. Thus, it may, for example, be used as an educational tool by pupils, students, and teachers or by artists, screenwriters, and social psychologists in order to exploit their expert knowledge in the respective areas. It is suited for the rapid prototyping of simple as well as the creation of sophisticated but still manageable models. Exploiting the modeling principles of modularity and compositionality, it allows the iterative and distributed development which reduces the modeling effort and complexity while improving maintainability and reusability.

ILLUSTRATION — MODELING SOCIAL AND COLLABORATIVE JOINT ACTIVITIES

In Chapter 5, I presented the conceptual design of the *BFML* ensemble members. This mainly aimed at introducing the formal foundations of the underlying modeling formalisms and giving an impression of the expressiveness and practicability of the modeling approach in view of the challenges described in Chapter 3. However, the chosen examples and illustrations remained rather basic and could be insufficient for fully understanding how the modeling concepts and languages are combined to a sophisticated behavior and interaction model that coordinates an agent’s behavioral aspects of interpersonal coordination and grounding.

For fully understanding the modeling approach, we now put hands on and explain how it is used in the development of a behavior and interaction model for an application with a joint activity between the user and a social agent. The therefore created model is designed to be pretty generic, that means it can be reused as whole model or in selected parts and adapted to specific concerns in similar applications. It served as tool chest for various demonstrator applications (Damian *et al.*, 2013; Baur *et al.*, 2013a; Mehlmann *et al.*, 2014b,a; Damian *et al.*, 2015; Mehlmann *et al.*, 2016; Wanner *et al.*, 2016) which were used to research the functions of gaze as well as the effect of turn-taking and interruption strategies on interpersonal coordination and grounding (Mehlmann *et al.*, 2014a,b; Gebhard *et al.*, 2014; Wanner *et al.*, 2016; Gebhard *et al.*, 2017) in social interactions with artificially intelligent agents.

In the remainder of this chapter, in Section 6.1, I first present the sensor setup, system, and software architecture as well as the scenario of the illustrative application and the underlying demonstrators. In Section 6.2, I explain how the user’s input events are modality-specifically preprocessed and how application-specific domain knowledge is represented. Finally, in Section 6.3, I provide a detailed description of the behavior flows that make up the central part of the agent’s behavior and interaction model. For readability and consistency with the previous chapters, these are depicted in their conceptual notation. The full model in its executable syntax, together with the *SWI-PROLOG* source code of all therein used *BFQL* predicates, is provided with the downloadable open-source version of the *VSM*³ authoring software¹.

¹<http://scenemaker.dfki.de>

6.1 Setup, Architecture and Scenarios

The behavior and interaction model, used for the illustration of the modeling approach in this chapter, is fairly powerful since it coordinates a lot of behavioral processes in a single computational model. It has emerged from multiple individual computational models each of which has been developed for a specific demonstrator application. The sensor and application setup, system and software architecture as well as the interaction scenarios of these demonstrators share a lot of characteristics that are also presupposed for the illustrative application in this chapter. The user interacts with a social robot companion or embodied conversational virtual character in a dyadic interaction, for example, by jointly playing a dialog-based serious game or cooperating in the completion of a collaborative joint activity.

6.1.1 System Setup And Architecture

APPLICATION &
SENSOR SETUP

Figure 6.1.1 shows the general application and sensor setup which is common to all demonstrator applications developed in the course of this thesis and underlying the illustrative application. The user and the agent are communicating via natural language and a variety of additional social signals and applications-specific actions in different modalities (Figure 6.1.1 (A)). Among the therefore used sensors and interaction devices are, for example, a *Tascam*² audio interface with a noise canceling microphone, *SMI*³ eye-tracking glasses, a *Microsoft®Kinect*⁴ sensor or a *Microsoft®Surface*⁵ table.



Figure 6.1.1: The general system setup of the applications that have been developed in this thesis.

The largest part of the user's multi-modal input is captured, synchronized, preprocessed, and interpreted using the *SSI* framework (Wagner *et al.*, 2013) (Figure 6.1.1 (B)). *SSI* allows configuring pipelines which execute sequences of preprocessing computations and data-driven or rule-based classification algorithms to interpret the user's input. Frequently used meth-

²<http://www.tascam.eu/>

³<http://www.smivision.com/>

⁴<http://www.kinectforwindows.org/>

⁵<http://www.pixelsense.com/>

ods are, for example, *SRGS*⁶ specifications, the *W3C* standard for speech recognition and parsing grammars, finite-state-based methods for gesture and posture recognition (Kistler, 2016), or hybrid probabilistic models with *Dynamic Bayesian Networks* (Murphy, 2002) for the recognition of higher level social attitudes and regulation strategies (Baur et al., 2015).

The recognized events are forwarded to the *VSM*³ framework (Figure 6.1.1 ©) where they are inserted to the event history of the *PROLOG* fact base and then further processed by the behavior and interaction model using the *BFQL*. The agents' interactive behavior is controlled with *scene flows*, the reference implementation of behavior flows in the *VSM*³ authoring suite (Gebhard et al., 2012; Mehlmann et al., 2016). Among others, the agents have been embodied conversational *Charamel CharActor*⁷ characters within a virtual *TriCat Spaces*⁸ environment or virtual characters in the *Horde 3D Game Engine*⁹. In other applications we used humanoid *NAO*¹⁰ or *Robokind*¹¹ robots as well as *Reeti*¹² or *Baxter*¹³ robots (Figure 6.1.1 ©).

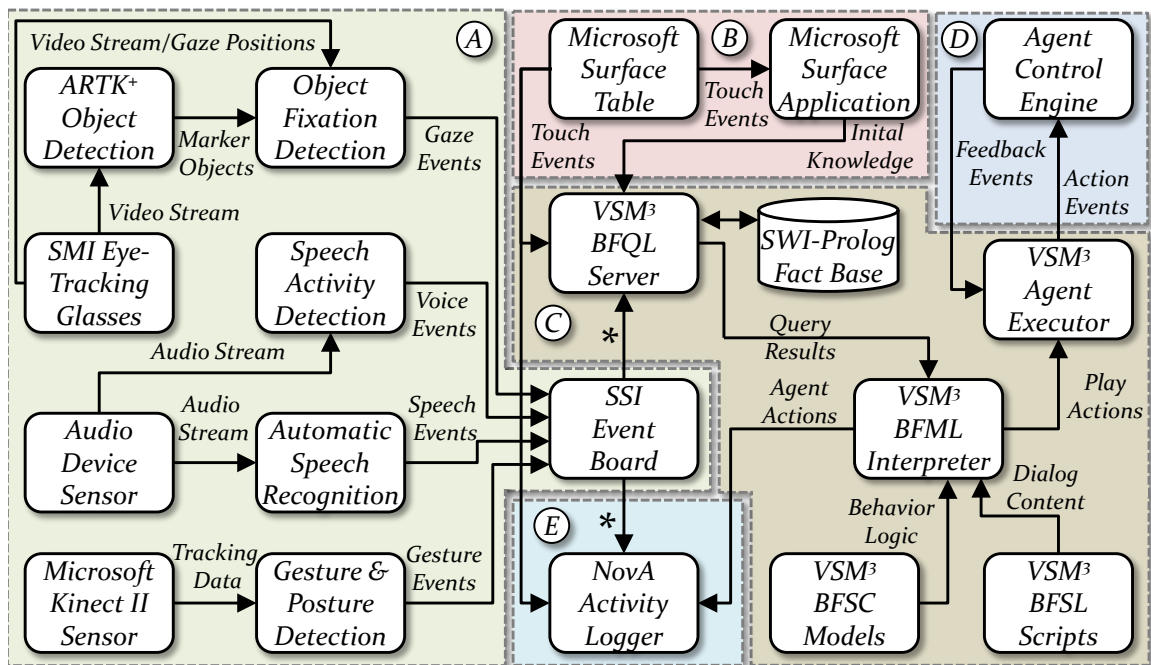


Figure 6.1.2: A schematic illustration of the data flow within our application's software architecture.

Figure 6.1.2 shows an illustration of the data and control flow between some important components of our illustrative application's software architecture. It depicts important SSI plugins and the data flow between them in the SSI framework (Figure 6.1.2 ©). Among those are, the gaze recognition pipeline with the *SMI* eye-tracking glasses, the *ARTK*¹⁴ marker detec-

SYSTEM &
SOFTWARE
ARCHITECTURE

⁶<https://www.w3.org/TR/speech-grammar/>

⁷<http://www.charamel.de/>

⁸<http://www.tricat.net/>

⁹<https://www.hcm-lab.de/>

¹⁰<http://www.aldebaran-robotics.com/>

¹¹<http://robokind.com/>

¹²<http://www.robopec.com/>

¹³<http://www.rethinkrobotics.com/>

¹⁴<http://www.artoolkit.org/>

tion and the object fixation recognition. Furthermore, it shows the voice activity and speech recognition pipelines based on the audio device sensor and the *Microsoft*[®] speech platform¹⁵ plug-in. It also contains the *Microsoft*[®]*Kinect*-based gesture, posture, and body property detection based on the *FUBI* (Kistler, 2016) and *NOVA* (Baur et al., 2015) plug-ins.

The architecture also shows the role of the surface table and the thereon executed application (Figure 6.1.2 ②). Other central components are the modules and knowledge bases of the *VSM*³ runtime environment (Figure 6.1.2 ③), including the *PROLOG* fact base with the *BFQL* server as well as the *BFSC* models and *BFSL* scripts. The diagram also shows how the agent platforms are integrated using platform-specific executor plug-ins that operate on the platforms' programming interfaces (Figure 6.1.2 ④). Finally, the architecture also comprises the *NOVA* activity logger (Figure 6.1.2 ⑤) which is used to automatically record the agent's and user's behavior for a later analysis and cooperative annotation (Baur et al., 2013b).

6.1.2 Interaction Scenario Description

ROBOTPUZZLE APPLICATION

The first application serving as role model for our illustrative application is the *ROBOTPUZZLE* application (Mehlmann et al., 2014b,a, 2016), a collaborative joint activity between the user and a social robot on a shared workspace. The robot companion is assisting the user during the completion of a sorting task on a surface table. In this, it is giving sorting instructions that can contain ambiguous verbal references to pieces and slots on the table. The user can consider the robot's directed gaze and pointing gestures or engage into a clarification dialog to resolve these ambiguities. When asking for clarification, the user may himself use ambiguous referring expressions, such that the robot can, the other way round, take the user's gaze into account for the multi-modal disambiguation of these references. The *ROBOTPUZZLE* was used to investigate the different functions of gaze behavior, especially joint attention, turn-taking and multi-modal disambiguation, for interpersonal coordination and grounding during such collaborative joint activities. The aim was to explore to which extent the robot's capability to master and coordinate the different social and functional aspects of gaze contributes to the effectiveness of the sorting task and his social perception.

The *SSI* pipeline in the *ROBOTPUZZLE* is used to interpret the user's gaze fixations to objects and areas on the table and to parse his dialog acts from clarification questions. The used robot is able to use natural language instructions that can be accompanied with pointing gestures and directed gaze to the objects and areas. The *VSM*³ authoring suite is used to model the dialog and interaction behavior of the robot instructor and, in particular, to realize the multi-modal disambiguation using the logic quantification of the *BFQL* and manage the robot's domain knowledge within the logic *PROLOG* fact base.

Figure 6.1.3 shows the demonstrator setup and sensor setting of the *ROBOTPUZZLE*. The user (Figure 6.1.3 ①) is sitting vis-à-vis a *NAO* robot (Figure 6.1.3 ②) on a *Microsoft*[®]*Surface* table (Figure 6.1.3 ③). The user wears *SMI* eye-tracking glasses and a noise canceling microphone for speech recognition (Figure 6.1.3 ④). A *Microsoft*[®]*Kinect* sensor is positioned behind the

¹⁵<http://msdn.microsoft.com/library/hh361572.aspx>

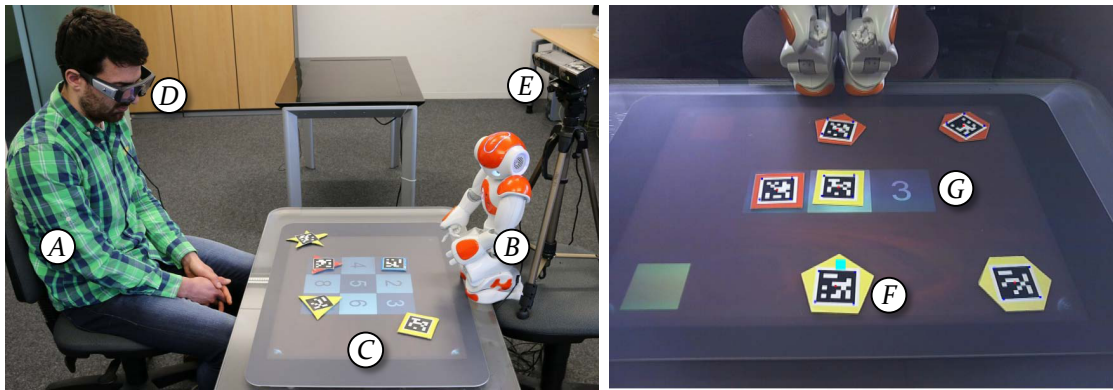


Figure 6.1.3: The scenario and setup of the *ROBOTPUZZLE* application on the left and a capture of the eye-tracking video that has been processed in the *SSI* framework (Wagner *et al.*, 2013) on the right.

robot for the recognition of the user’s head movements and gestures (Figure 6.1.3 (E)). The puzzle pieces on the surface table have distinguishable features such as a size, color, shape and position. They are marked with *Byte Tags* on their bottom side in order to track their position and with *ARTK⁺* markers on their top side to recognize the pieces the user is looking at via marker tracking on the video of the eye-tracking glasses (Figure 6.1.3 (F)). The pieces have to be sorted into different puzzle slots that are realized with overlay areas which are displayed on the surface table and are labeled with unique numbers (Figure 6.1.3 (G)).

The interaction scenario of the illustrative application corresponds to the largest part to that of the *ROBOTPUZZLE* but shares additional characteristics with the *SOCIALCOACH* application (Damian *et al.*, 2013; Baur *et al.*, 2013a; Damian *et al.*, 2015). This application realizes a kind of serious game between the user and various virtual social coach characters in form of a social job application training. The training comprises a simulated job interview in which the user is confronted with interview partners with varying personalities using different interview strategies. A consecutive debriefing phase is used to recap, discuss and assess the user’s behavior in specific interview situations. The *SOCIALCOACH* was developed with the goal to investigate different emotion regulation and coping strategies of users as well as the function of interruptions and the perception of the agent’s interruption handling strategies during this kind of social interaction. With regard to user interruptions, we wanted to explore to which extent different interruption handling strategies of the agent influence the assessment and perception of the agent’s dominance, involvement and friendliness as well as the comfortableness of the user.

SOCIALCOACH
APPLICATION

The *SSI* processing pipeline in the *SOCIALCOACH* is able to recognize the user’s social cues and behavioral patterns for coping strategies and emotion regulation using a hybrid, that means theory-based and data-driven approach (Baur *et al.*, 2015) as well as the user’s voice activity and spoken keywords with a simple threshold-based method. The character animation and virtual environment engine provides different sceneries and virtual embodied conversational characters. The *VSM³* authoring suite is used to model the dialog and interaction behavior of the different interview partners and debriefing assistants.

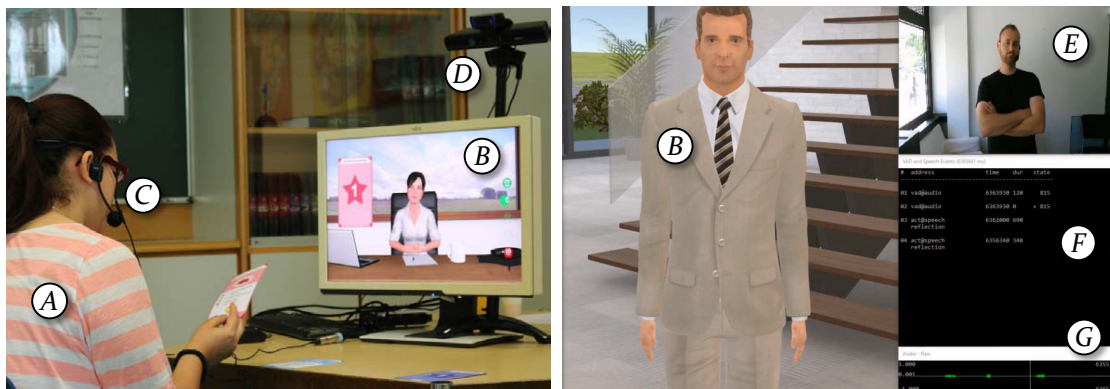


Figure 6.1.4: The setup of the *SOCIALCOACH* application with the user, the sensor setup, the virtual embodied characters as well as the SSI pipeline visualization and event monitor (Wagner *et al.*, 2013).

Figure 6.1.4 shows a demonstrator setup and sensor setting of the *SOCIALCOACH*. The scenario shows the human user (Figure 6.1.4 (A)) in front of a screen displaying an embodied conversational *Charamel CharActor* character within the virtual *TriCat Spaces* environment (Figure 6.1.4 (B)). The user wears a noise canceling microphone for voice activity and speech recognition (Figure 6.1.4 (C)). A *Microsoft®Kinect* sensor is positioned behind the screen for the recognition of the user’s head movements, gestures and body properties (Figure 6.1.4 (D)). The SSI framework (Wagner *et al.*, 2013) captures and synchronizes the user’s skeleton data and camera image (Figure 6.1.4 (E)) and reports voice activity and keyword events (Figure 6.1.4 (F)) based on the user’s preprocessed audio input (Figure 6.1.4 (C)).

SCENARIO OF
ILLUSTRATIVE
APPLICATION

The interaction scenario of the illustrative application combines those of the *ROBOTPuzzle* and the *SOCIALCOACH* application. In addition, the *Microsoft®Kinect* is used to detect facial expressions using an SSI module based on the *SHORE*¹⁶ or *OPENFACE*¹⁷ libraries. Thus, the agent’s behavior comes very close to that of the social robot Charly in the example from Section 1.2. Resembling human interaction, both partners exploit various multi-modal behaviors for interpersonal coordination and grounding. Gaze cues are aligned with verbal contributions and touch actions to regulate the speaker and listener roles and ensure a fluent interaction flow. Touch and speech behaviors are used for interruption attempts that are handled based on parameterizable detection and handling policies. Gaze is used to continually give and elicit feedback signals and follow or direct the other’s attention to objects, events or persons. The combination of gaze, gestures and speech is used to multi-modally refer to objects on the workspace. Disruptions of the common ground due to ambiguous referring expressions are disambiguated by considering the partner’s gaze. Unresolved ambiguous references can be disambiguated by engaging in a clarification dialog. The illustrations in the following sections illustrate the coordination of these behavioral functions and their integration with the floor and dialog management in a single behavior and interaction model.

¹⁶<https://www.iis.fraunhofer.de/>

¹⁷<https://cmusatyalab.github.io/openface/>

6.2 Preprocessing Input and Context

The user's voice, speech, gaze, and all other input events, except touch events, in the illustrative application are captured, synchronized, preprocessed, and interpreted by the respective modality-specific interpretation plug-ins of the *SSI* framework. Touch events, such as the movement, placement, and dragging of puzzle pieces on the surface table are directly forwarded from by the surface application to *VSM*³. The corresponding event feature structures are asserted to the event history of the logic fact base. They carry modality-independent features, such as timestamps and confidence values, and modality-specific semantic information, such as gaze target distributions and abstract dialog acts. The application knowledge is initialized by the surface application, then transmitted to the *VSM*³ and as well represented as feature structures in another part of the logic fact base. It comprises information about the individual puzzle slots and pieces as well as the robot's instructions for a sorting task.

6.2.1 Representing Domain Knowledge

Figure 6.2.1 shows feature structures representing the knowledge about an exemplary puzzle piece (Figure 6.2.1 (A)) and a particular puzzle field (Figure 6.2.1 (B)). Among other attributes, they contain unique name and unambiguous description which are used to unambiguously refer to them in instructions or clarifications. A piece additionally contains distinguishable features, such as a size, color, and shape as well as a position and a state that encode whether and where it is lying on the table. A field contains a unique number which is displayed on its overlay on the surface as well as a position and bounding area which are used to check if a puzzle piece has been placed on it.

REPRESENTING
FIELDS & PIECES

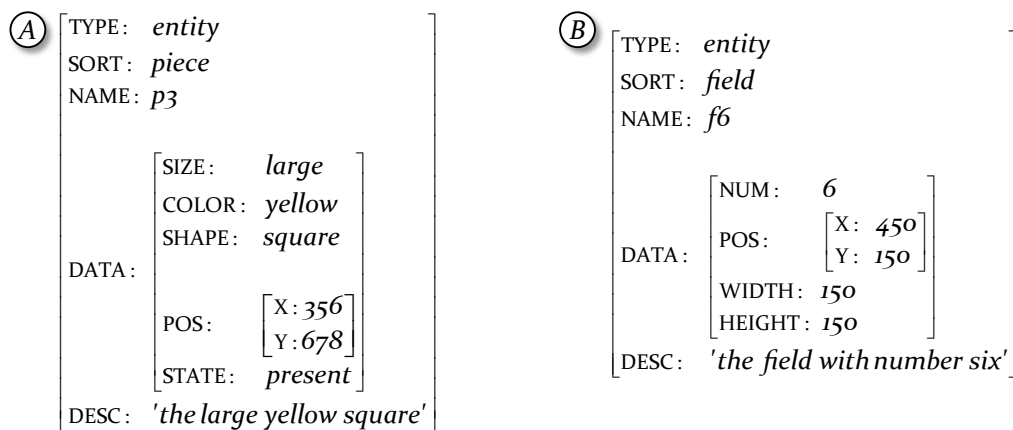


Figure 6.2.1: The exemplary representation of a puzzle piece (A) and a field (B) as feature structures.

Figure 6.2.2 shows two feature structures representing exemplary sorting task instructions. They contain the information needed by the agent to give the respective instruction and check if the user properly executed it. This include a unique name and possibly ambiguous textual description that is used for the instruction. For example, the instruction “*move the green puzzle piece to the field with number six*” (Figure 6.2.2 (A)) could be ambiguous if there would lie two or more green puzzle pieces on the surface table. The instruction “*move the*

REPRESENTING
INSTRUCTIONS

small red star to the field with number seven” (Figure 6.2.2 ②) is, in turn, accurately specified because it specifies all three distinguishable features and can not be ambiguous because there can only exist one single puzzle piece with exactly the same feature set. An instruction also contains the identifiers of the puzzle piece and field used in this instructions. These can be used to check if the user properly understood the instruction and moved the correct puzzle piece to the intended puzzle field. They can also be used to construct unambiguous descriptions of the instruction for a clarification statement by falling back on the aforementioned unambiguous denominations of the corresponding entities.

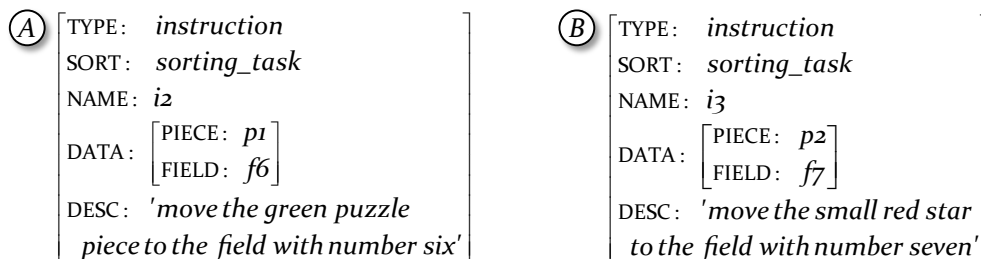


Figure 6.2.2: The exemplary representation of an ambiguous ① and an unambiguous instruction ②.

6.2.2 Preprocessing User Input Events

Figure 6.2.3 shows some examples of feature structures for rather simple event types that carry only lexical content. The first two represent state events (Figure 6.2.3 ①) which inform the model if user and agent are taking part in the interaction. They must be produced by an *SSI* module which signals whenever a participant is entering or leaving the interaction, for example, based on a suitable visual sensor device. The next two feature structures represent voice events (Figure 6.2.3 ②) which notify the model whenever the user or the agent start or stop speaking. The user’s voice events are produced by the voice activity recognition module listening on a microphone. The agent’s voice events must be produced by the agent’s executor instance when receiving speech synthesis notifications from the agent’s text-to-speech engine. The last structure represents a facial expression event (Figure 6.2.3 ③) which is produced by an appropriate facial expression recognition module of *SSI* whenever the user displays an emotion or a cognitive state.

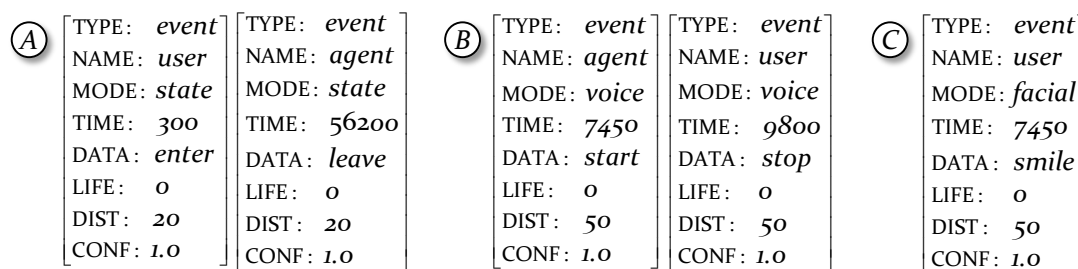


Figure 6.2.3: Examples of simple input events, like state ①, voice ② and facial expression events ③.

Figure 6.2.4 shows different touch events that are produced by the surface application when the user starts (Figure 6.2.4 ①), continues (Figure 6.2.4 ②) or stops dragging a puzzle piece. They are similar to object movement events, which are generated by the surface table whenever the user lays down a puzzle piece or has just finished moving it to a field or position on the surface. A touch event therefore has a type which specifies if the user just starts, stops, or continues dragging as well as the name and current position of the moved object.

*SURFACE TABLE
OBJECT TOUCH
& MOVE EVENTS*

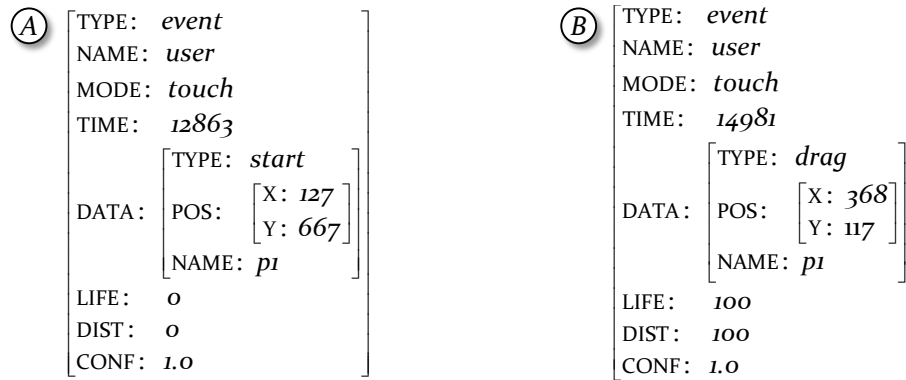


Figure 6.2.4: The exemplary representation of a start touch ① and a drag continue touch event ②.

Even the most accurate eye-tracking system has to cope with noise during the measurement of gaze fixations and eye movements. A certain degree of noise is introduced by the technology-related inaccuracy due to the imperfectness of the calibration-based transformation algorithms as well as sensor device errors and loss of data. The larger proportion is the result of biology-related factors due to eye movements, such as tremor and micro-saccades, eye jittering, eye blinking and also a natural random offset due to fuzzy fovea dimensions between the vector of actual attentive gaze direction and eye optical axis (Špakov, 2011).

*EYE-TRACKING
NOISE &
JITTERING*

When using a mobile eye-tracker and physical objects, as in the illustrative application, the fixated objects on a surface are determined by mapping gaze points to recognized objects on a constantly changing video image. This association of gaze points with objects is substantially influenced by the quality of the computer vision methods for object or marker recognition on the video image (Freeman *et al.*, 2007). Errors in detecting the visual parameters, such as thermal noise, pixelization, or calibration, and algorithmic errors, changing light conditions and different viewing angles, can cause object recognition errors that sum up with the noise of the eye-tracking device to false measurements.

To cope with such noise, the recognition module must average the user's gaze target mapping over several frames. Therefore, it first selects the object which is geometrically including the gaze point (Hansen *et al.*, 2001), is the closest of all objects to it (Monden *et al.*, 2005), or is among a few nearest objects to it (Xu *et al.*, 2008). Then it compensates for object tracking errors or gaze point losses by applying a smoothing algorithm over several frames. This helps to transform the noisy raw gaze and object recognition data into a smooth and coherent object gaze mapping or probability distribution (Mehlmann *et al.*, 2014a).

*GAZE DATA
PREPROCESSING
ALGORITHM*

Algorithm 6.2.1 The algorithm that computes regular gaze events from the raw video image and coordinate streams provided by the eye-tracking glasses and the list of ARTK⁺ markers.

```

1: procedure TRANSFORM(Video v, Point g, List l)
2:    $\Delta_{max} \leftarrow \sqrt{v_w^2 + v_h^2}$ 
3:   if  $g_x \geq 0 \wedge g_x \leq v_w \wedge g_y \geq 0 \wedge g_y \leq v_h$  then
4:     for all  $m \in l$  do
5:       if visible( $m$ ) then
6:          $\Delta_m \leftarrow \sqrt{(g_x - m_x)^2 + (g_y - m_y)^2}$ 
7:       else
8:          $\Delta_m \leftarrow \min(\delta \cdot \Delta_m, \Delta_{max})$ 
9:       end if
10:       $\Phi_m^* \leftarrow (1 - (\Delta_m / \Delta_{max}))^\varphi$ 
11:       $\Phi_m \leftarrow \sigma \cdot \Phi_m + (1 - \sigma)\Phi_m^*$ 
12:    end for
13:  end if
14: end procedure

```

For recognizing the user’s gaze targets, the *SSI* framework was extended with a plug-in that can work with raw noisy eye-tracking and marker-tracking data. It implements the simple algorithm shown in Algorithm 6.2.1 to reduce the influence of recognition errors, data losses or outliers that occur, for example, whenever the user blinks, rolls his eyes, or is shortly distracted. The algorithm also reduces the sample rate such that the gaze events arrive at *VSM*³ with a lower customizable frequency. In each frame, the algorithm computes the distances of all recognized puzzle pieces and the robot’s chest to the user’s gaze position (Algorithm 6.2.1 ⑥). Then it computes a fixation confidence for each marker based on this distance and the respective confidences in the past few frames (Algorithm 6.2.1 ⑩).

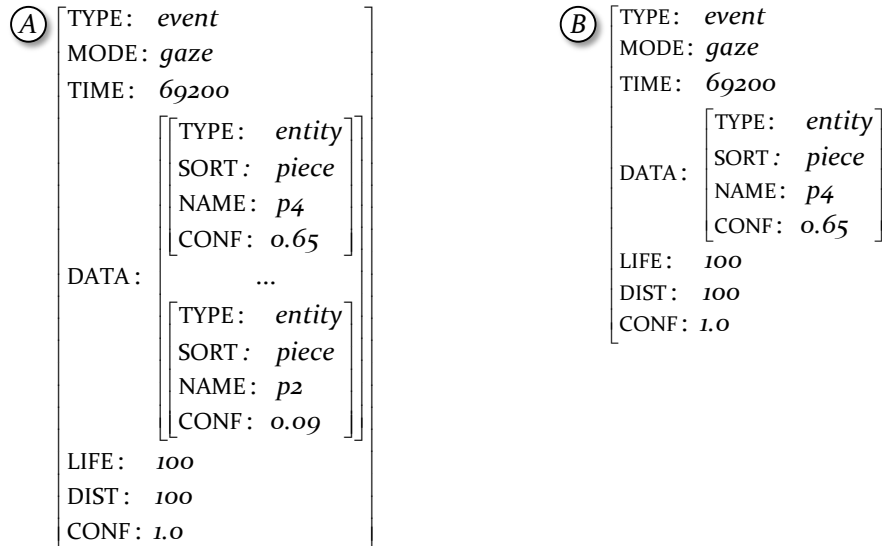


Figure 6.2.5: The exemplary representation of a gaze distribution ① and a gaze target event ②.

pothesis every couple of frames. As shown in Figure 6.2.5, a gaze distribution event contains a list with an entries for each individual puzzle piece containing the probability that the user looks at this specific puzzle piece (Figure 6.2.5 ①). As simplification, but without restriction of generality, the illustrative model uses simple gaze target events containing only the puzzle piece with the highest probability, so to say the best hypothesis (Figure 6.2.5 ②).

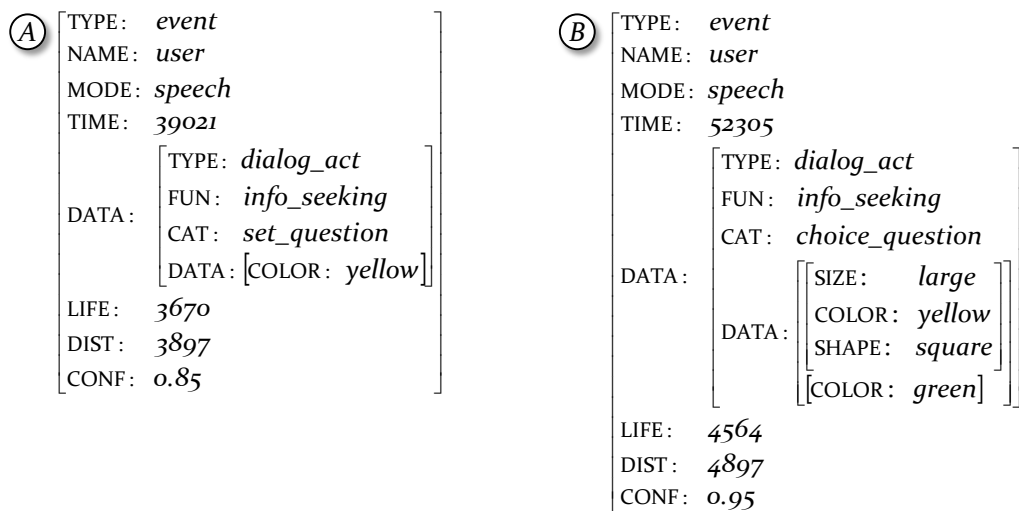


Figure 6.2.6: The exemplary representation of a set ① and a choice question ② speech act.

Figures 6.2.6 and 6.2.7 show examples of the different types of speech events that are produced by the natural language understanding pipeline of the SSI framework. In this, the user’s utterances are transcribed and parsed into abstract dialog acts using a semantic parser which is based on the Microsoft® speech platform. According to the DiAML¹⁸ classification scheme (Bunt *et al.*, 2010; Bunt, 2011; Bunt *et al.*, 2012), the *communicative function* of the user’s clarification questions is *information seeking*. Based on the type of question the resulting dialog act can have three different semantic categories. A *set question* asks for a decision between an unspecified number of entities that fit to a description that the robot used in the preceding instruction. They are usually used to ask for the refinement of a description in order to narrow down the list of possible alternatives. Figure 6.2.6 ① shows a feature structure representing the set question “*which yellow one do you mean?*” which could be asked in reaction to the agent’s ambiguous instruction to move a yellow puzzle piece. A *choice question* asks for the decision for a specific puzzle piece from a list of possible candidates. Figure 6.2.6 ② shows a feature structure representing the choice question “*do you mean the green one or the large yellow square?*”. A *check question* is a type of *propositional question* that is used by the user to check the understanding of an entity’s description. They are usually used to get a confirmation that the agent actually proposed or referred to a particular puzzle piece. Figure 6.2.7 ① shows a feature structure representing the check question “*do you mean the large yellow square?*” used as reaction to the agent’s instruction to move a square piece. In the *ROBOTPUZZLE* application, a specific application-specific part of the logic is used to resolve unimodal referring expressions such as “*the yellow square*” by implementing an algorithm

SPEECH EVENTS
& DIALOG ACTS

¹⁸<http://semantic-annotation.uvt.nl>

similar to that described by [Ros et al. \(2010\)](#). It determines a set of puzzle pieces that match the list of features from the user’s description and computes the optimal set of discriminating features to correct the user with an unambiguous answer.

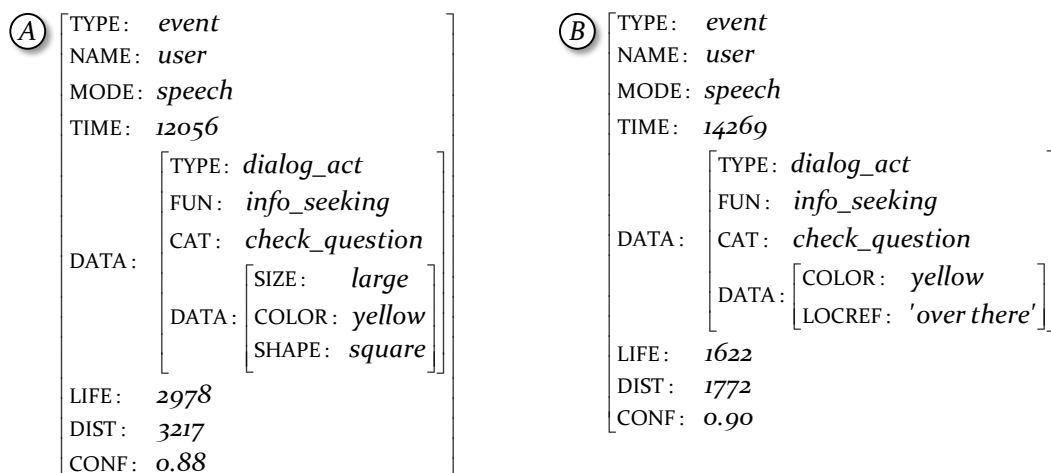


Figure 6.2.7: The exemplary representation of a check question without ① and with spatial deixis ②.

6.2.3 Disambiguating Speech with Gaze

All types of questions that the user might ask can contain verbal referring, in better words, spatial deictic expressions ([Levinson, 2008](#)), such as “*there*” or “*here*”. For example, [Figure 6.2.7 ②](#) shows the representation of the check question “*do you mean the yellow one over there?*” which contains the spatial deictic referring expression “*over there*”. The user would usually use such a formulation when referring to a specific puzzle piece while pointing or looking at it. Consequently, if the user’s utterance contains a location referent then the agent should consider his referential gaze or directed pointing gestures to resolve the multi-modal reference. The *BFQL* provides several predicates to disambiguate the user’s verbal references with his gaze fixations during a surrounding time window. These rely on the predefined *BFQL* predicates evaluating *temporal constraints* and *generalized quantifiers* as well as application-specific *matching predicates* examining the *semantic accordance* of the user’s verbal feature descriptions with the features of all available puzzle pieces.

[Listing 6.2.1](#) shows the Prolog implementation of a predicate that tries to disambiguate the location referent carried by the dialog act of the user’s check questions with the user’s gaze. It first checks if the dialog act of the speech event *Speech* represents a check question and contains a location referent at all. In this case, it uses the generalized quantifier *forlargest/3* to find the name of the puzzle piece that has been looked at the largest amount of time during the speech event and whose attributes match the features description that the user provided with its check question. If this can be found, then it constructs a new event *Fused* from the original speech event *Speech* that additionally contains a new feature *DATA:DATA:NAME* holding the name of the found puzzle piece.

```

disambiguate(Speech, Fused) :-
    val(data:fun, info_seeking, Speech),
    val(data:cat, check_question, Speech),
    val(data:data:locref, _, Speech),
    foralllargest(Gaze, (fsr(Gaze), /* Quantification */
        val(mode, gaze, Gaze), /* Type constraint */
        during(Gaze, Speech), /* Temporal constraint */
        matches(Gaze, Speech)), /* Semantic constraint */
    val(data:name, Name, Gaze)), !,
    set(data:data:name, Name, Speech, Fused).
disambiguate(Speech, Fused) :-
    val(data:fun, info_seeking, Speech),
    val(data:cat, check_question, Speech),
    findall(Name, (fsr(Piece), /* Alternative collection */
        val(sort, piece, Piece), /* Type constraint */
        matches(Piece, Speech), /* Semantic constraint */
    val(name, Name, Piece)), List),
    set(data:data:name, List, Speech, Fused).

```

Listing 6.2.1: The SWI-PROLOG implementation of the disambiguation predicate *disambiguate/2*.

Otherwise, if the evaluation of the generalized quantifier and, thus, the search for the puzzle piece fails, then the predicate collects all puzzle pieces that match the description in the question into a list of possible candidates. Afterwards, it constructs the new fused event *Fused* like above, with the exception that the feature `DATA:DATA:NAME` holds this candidate list. Thus, the reference remains unresolved if the list contains more than one alternative. This situation must then be handled by the dialog manager by engaging in a clarification dialog, a topic out of the scope of this thesis. A variation of the disambiguation predicate could use another generalized quantifier or an adapted generator condition. It could, for example, only choose gaze events with a certain minimum confidence value or consider gaze events in another time window than exactly during the speech event.

Figure 6.2.8 shows an illustration how the *foralllargest/3* quantifier supports the disambiguation in the just mentioned case. In this example, the user asks the ambiguous check question “*The green one over there?*” (Figure 6.2.8 (B)) while looking at three different puzzle pieces on the surface table, two of which are green and one is red (Figure 6.2.8 (A)). The application of the generalized quantifier first finds all gaze events during the user’s utterance (Figure 6.2.8 (C)). Then it chooses those whose gaze target’s attributes match with the feature description in the check question, basically all green puzzle pieces during the utterance (Figure 6.2.8 (D)). The *foralllargest/3* quantifier then chooses the set of gaze events whose gaze targets additionally share the same name (Figure 6.2.8 (E)), that means refer to the same puzzle piece, and selects the one that is being looked at for the largest amount of time (Figure 6.2.8 (F)). As a result, the disambiguated multi-modal event carries the new unique name feature `DATA:DATA:NAME` with the value *p3* (Figure 6.2.8 (G)) and can then be propagated to the dialog management.

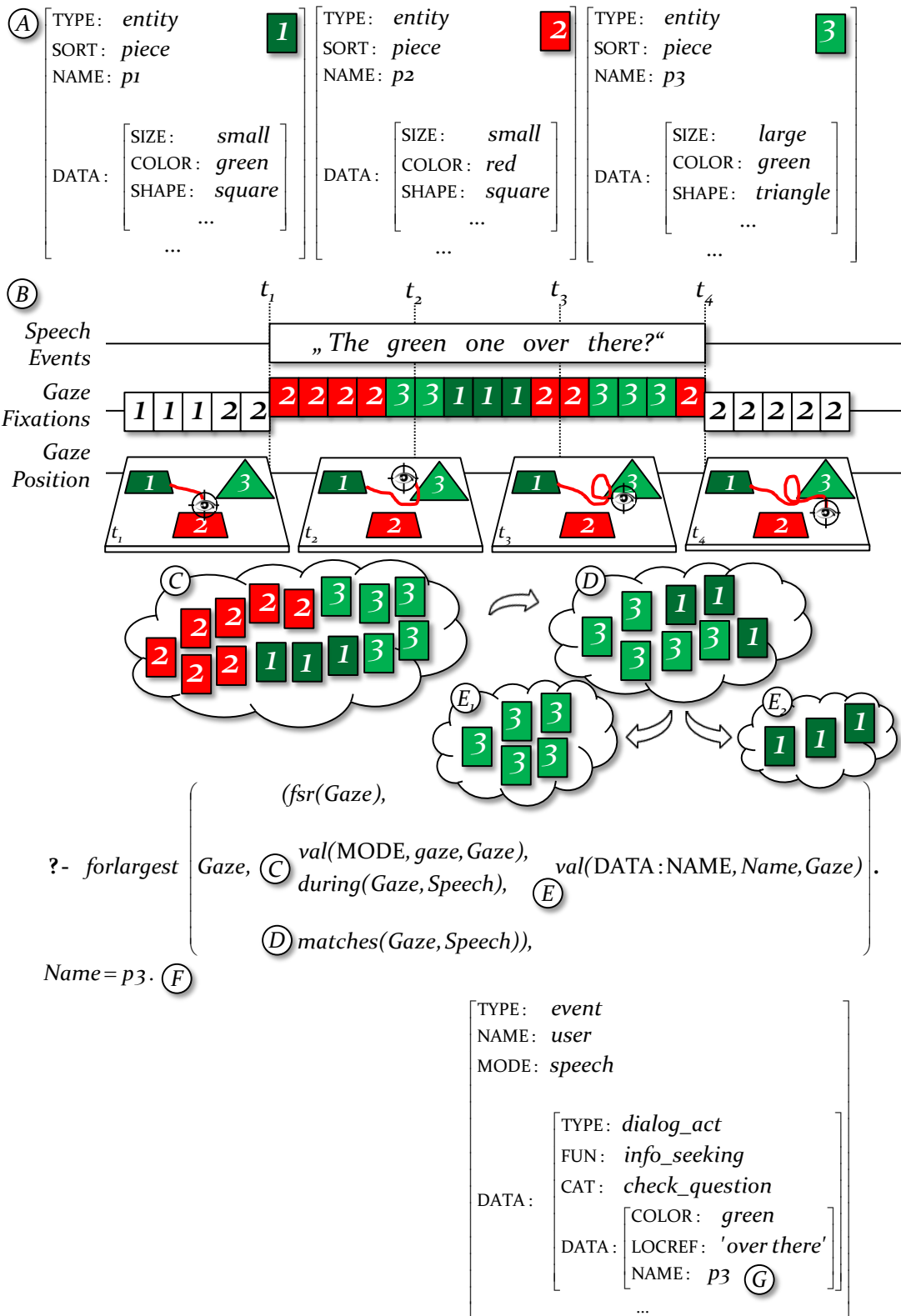


Figure 6.2.8: An illustration of how the *forlargest/3* quantifier supports the disambiguation.

6.3 Modeling Behavior and Interaction

Figure 6.3.1 shows the overall architecture of the behavior flow which models the agent's behavior and interaction in the illustrative application. It makes extensive use of parallel and hierarchical decomposition comprising multiple concurrent and nested layers. The individual layers are building and heavily relying upon each other to implement various behavioral levels and functions. These are, among others, input detection and preprocessing, multi-modal integration and disambiguation, behavioral pattern recognition, participant role management, dialog and behavior control, and particular subtasks. They are coordinated through shared global variables and the exchange of events via the logic fact base. This hierarchical and parallel structure can easily be reused and adapted and, thus, be considered as best practice method to structure similar behavior and interaction models using behavior flows.

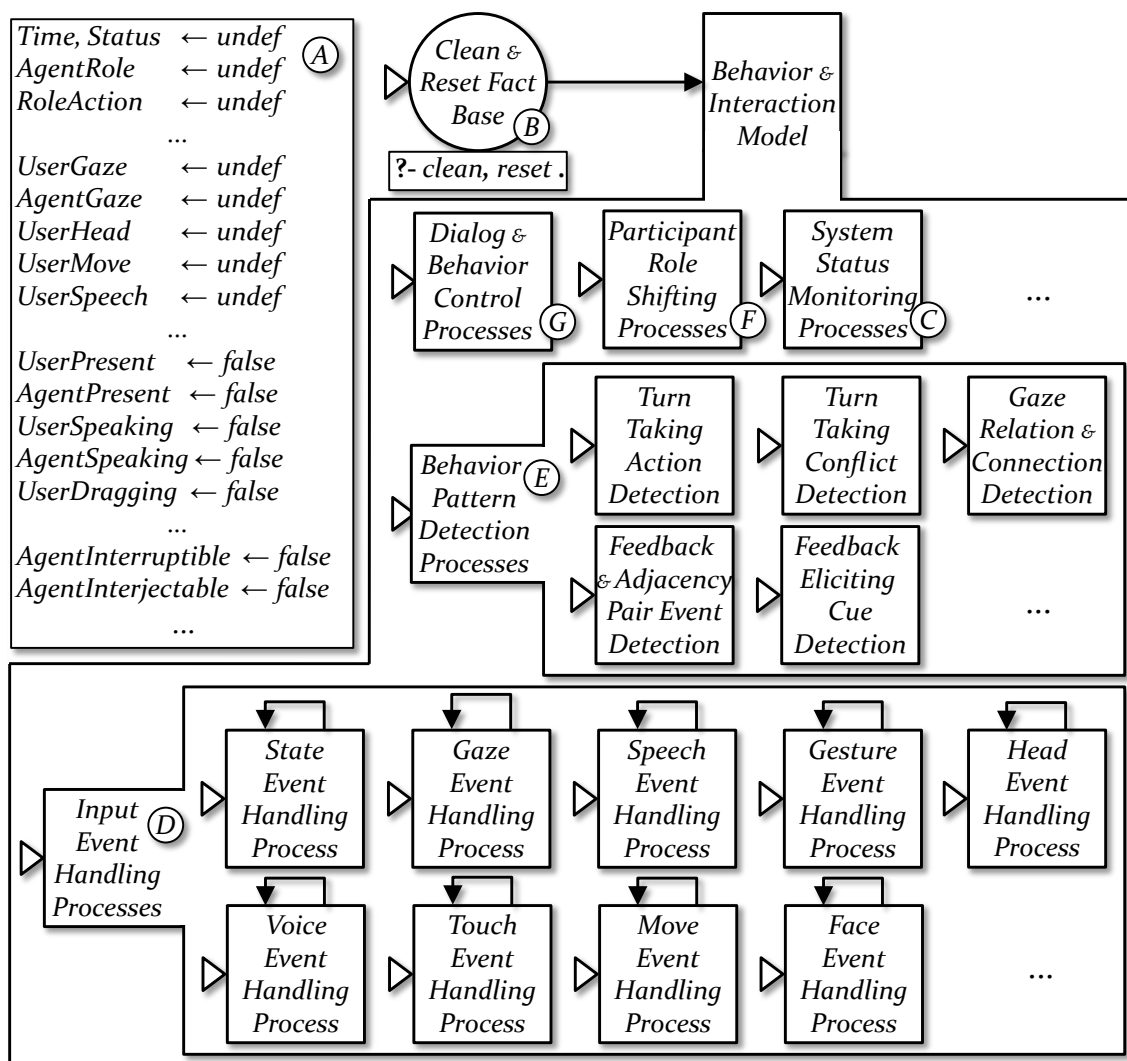


Figure 6.3.1: The overall architecture of the behavior flow model used in the illustrative application.

On the highest level, the behavior flow defines the global variables that are shared by all nested behavior flows (Figure 6.3.1 A). It starts with the call to the predicates *clean/o* and

reset/o (Figure 6.3.1 B) to clear the fact base from events, signals, and timers and reset the application-specific domain knowledge about entities and instructions of the sorting task. Afterwards, several nested behavior flows representing the agent's behavioral levels and processes are concurrently executed. Among those are a *system status monitoring layer* for monitoring the system statistics and managing the garbage collection (Figure 6.3.1 C), an *input event handling layer* for preprocessing the user's and agent's input events (Figure 6.3.1 D), a *behavioral pattern recognition layer* for the incremental and concurrent recognition of multi-modal and bidirectional behavioral patterns (Figure 6.3.1 E), a *participant role management layer* for the assignment and shifting of participant roles (Figure 6.3.1 F) as well as a *dialog and behavior control layer* for dialog flow management and role-specific aspects of the agent's nonverbal behavior (Figure 6.3.1 G). The following sections describe the most important layers and contained behavior flows in detail and explain how they are coordinated.

6.3.1 Basic Input Event Handling

The behavior flows on the *input event handling layer* (Figure 6.3.1 D) are preprocessing events from modality-specific recognition modules and application-specific devices. They constantly observe the logic fact base and extract required information from incoming event features structures using the operations defined in Section 5.3.3. On the one hand, they update particular global variables with this information (Figure 6.3.1 A), such as the participants' voice activities and gaze targets. On the other hand, the information may be combined and propagated with signals directed to individual or distributed to whole sets of behavior flows on higher layers (Figure 6.3.1 E, F, G). Due to space and redundancy reasons, we only explain some selected behavior flows of this layer which could easily be extended with additional parallel behavior flows for handling further input modalities and devices.

Handling Voice Events

The behavior flow in Figure 6.3.2 is handling the participants' *voice events*. The user's voice events are produced by the voice activity recognition (Figure 6.1.2 A) whenever he starts or stops speaking into a microphone. The agent's voice events must be produced by the agent's executor instance (Figure 6.1.2 C) whenever it receives speech synthesis notifications from the agent's text-to-speech engine (Figure 6.1.2 D). The behavior flow starts by waiting for the next voice event to occur in the fact base (Figure 6.3.2 A). It uses the predicate *voice/2* (Figure 6.3.2 B) to retract a new voice event from the fact base and store the event's name and data features in the local variables *Name* and *Data* (Figure 6.3.2 C). Then, it checks if the event has been caused by the user or the agent (Figure 6.3.2 D) and if the respective participant has just started or stopped speaking (Figure 6.3.2 E). Afterwards, it updates the global variable *UserSpeaking* or *AgentSpeaking* (Figure 6.3.2 F) to inform processes on higher layers about the voice activity change (Figure 6.3.2 G). Among those are the behavior flow recognizing turn-taking actions in Figure 6.3.6, the one detecting voice activity overlaps in Figure 6.3.8, and those detecting nonverbal back-channels and adjacency pairs in Figures 6.3.15 and 6.3.16. The behavior flow then restarts to handle more voice events (Figure 6.3.2 H).

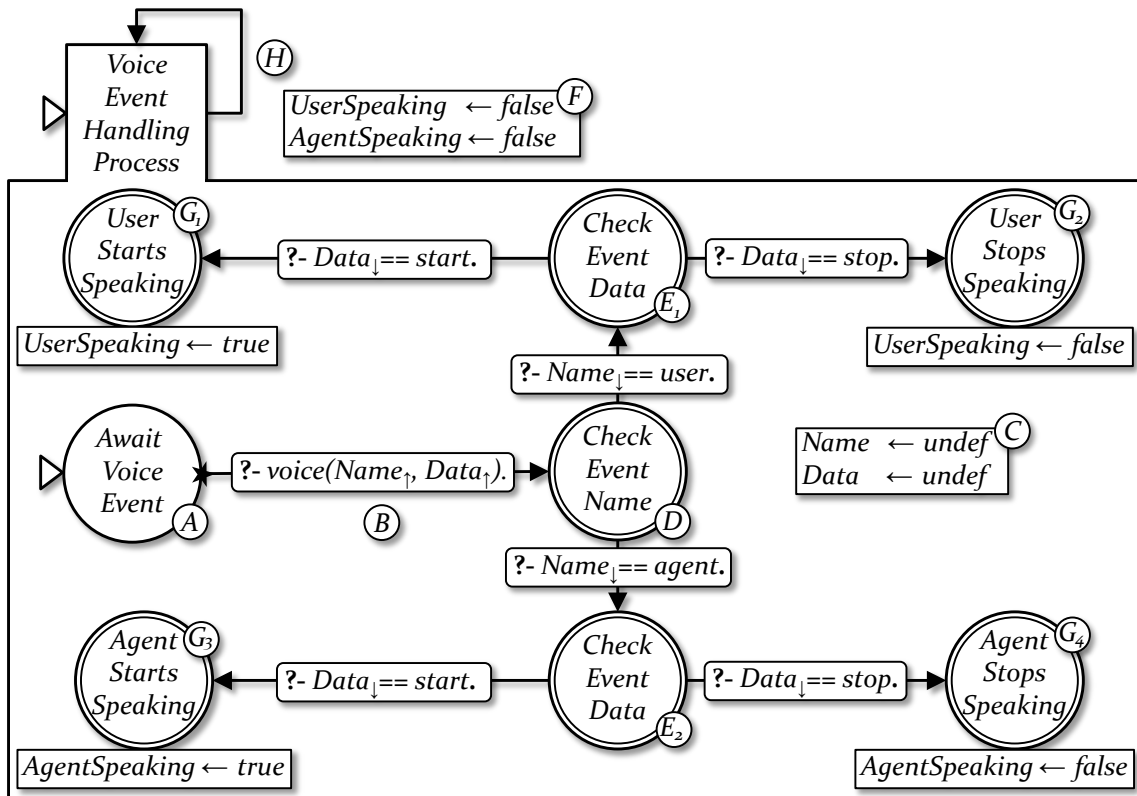


Figure 6.3.2: The behavior flow handling the user's and the agent's incoming voice activity events.

Handling Gaze Events

The behavior flow in Figure 6.3.3 is preprocessing the participants' *gaze events*. The user's gaze events are continuously provided by the gaze recognition module (Figure 6.1.2 (A)) based on the eye- and object-tracking data using Algorithm 6.2.1. The agent's gaze target is transmitted by the agent platform (Figure 6.1.2 (D)) in regular time intervals of a few milliseconds. The last processed gaze event is always stored in the local variable *Last* (Figure 6.3.3 (A)). The behavior flow starts by waiting for the next gaze event to arrive at the fact base (Figure 6.3.3 (B)). It uses the predicate *gaze/4* (Figure 6.3.3 (C)) to check if a younger gaze event exists and extract the event and its name and data features to the local variables *Last*, *Name* and *Data* (Figure 6.3.3 (A)). Gaze events are not retracted but kept in the event history since they are needed for the multi-modal disambiguation of verbal references in Figure 6.3.5, as explained in Section 6.2.3, and are later retracted by the garbage collection. Afterwards, the behavior flow first checks if the event belongs to the user or the agent (Figure 6.3.3 (D)) and if the participant's gaze target has changed since his last gaze event (Figure 6.3.3 (E)). In case of the agent, it produces a *gaze shift signal* (Figure 6.3.3 (F)) that is directed to behavior flow in Figure 6.3.12 which recognizes *gaze connection events*. If the user's gaze target has changed, then it produces a number of gaze signals directed to different behavior flows (Figure 6.3.3 (G)). The behavior flow in Figure 6.3.12, which recognizes *gaze connection events*, expects a *gaze shift signal* when the user's visual focus has changed. The process in Figure 6.3.26 depends on a *gaze focus signal* to let the agent follow the user's visual attention in the addressee role.

The behavior flow that recognizes *back-channel eliciting cues* in Figure 6.3.18 expects a *gaze glance signal* to work. Afterwards, the behavior flow updates the global variables *UserGaze* or *AgentGaze* (Figure 6.3.3 (H)) to inform other processes about the changing gaze targets (Figure 6.3.3 (I)) before it starts over again to handle the next gaze events (Figure 6.3.3 (J)).

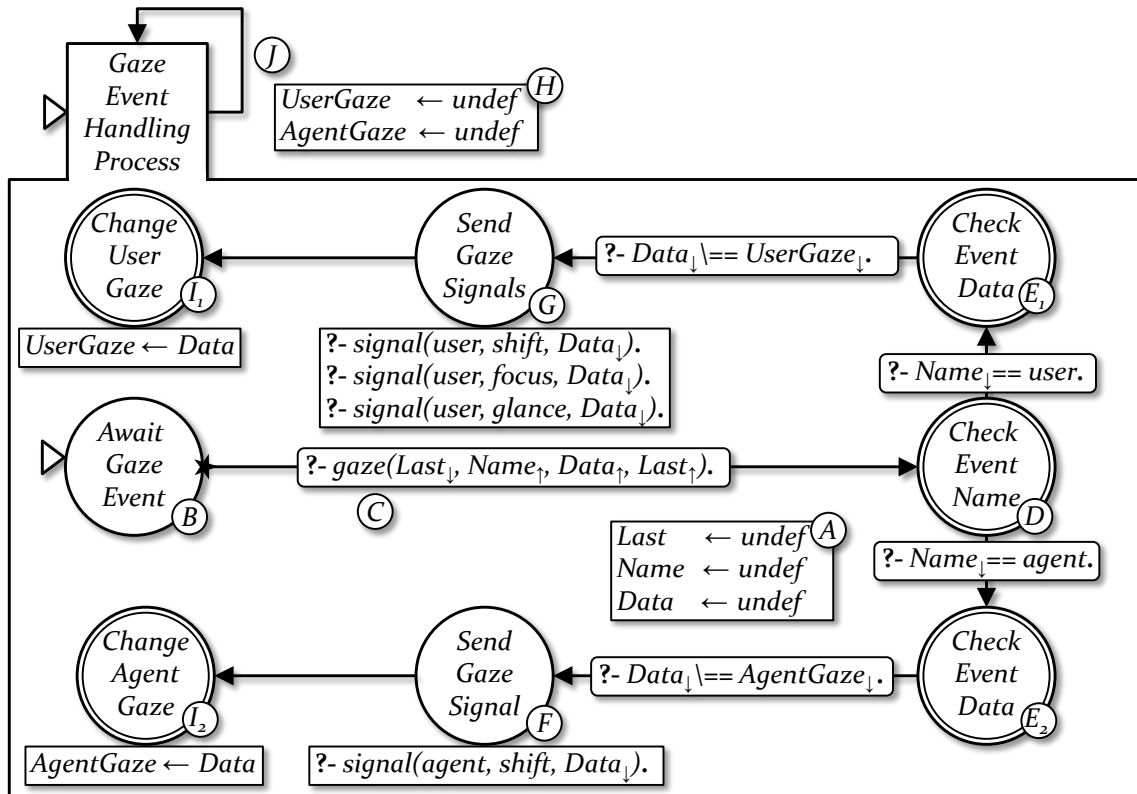


Figure 6.3.3: The behavior flow used for preprocessing the two participants' new gaze events.

Handling Touch Events

The behavior flow in Figure 6.3.4 is handling the user's *touch events* that are produced by the surface application (Figure 6.1.2 (B)) when the user starts, continues, or stops dragging a puzzle piece. It starts by waiting for the next touch event to arrive at the fact base (Figure 6.3.4 (A)). It uses the predicate *touch/4* (Figure 6.3.4 (B)) to retract a new touch event from the fact base and extract its type and name features as well as the position of the moved object to the local variables *Type*, *Name*, *X* and *Y* (Figure 6.3.4 (C)). Then, it updates the position knowledge in the logic fact base using the predicate *update/3* and inspects the type of the touch event (Figure 6.3.4 (D)). In case of a drag action, it produces a *drag action signal* (Figure 6.3.4 (E)) directed to the behavior flow in Figure 6.3.26 which coordinates the agent's gaze behavior in the addressee role. In case of a start action (Figure 6.3.4 (F)) or a stop action (Figure 6.3.4 (G)), it updates the global variable *UserDragging* (Figure 6.3.4 (H)) to notify the higher layers about the object movement. For example, the behavior flow recognizing turn-taking actions in Figure 6.3.6 is using this information to decide if the user wants to take, hold, yield, or assign the turn. Finally, the behavior flow restarts to handle the next touch events (Figure 6.3.4 (I)).

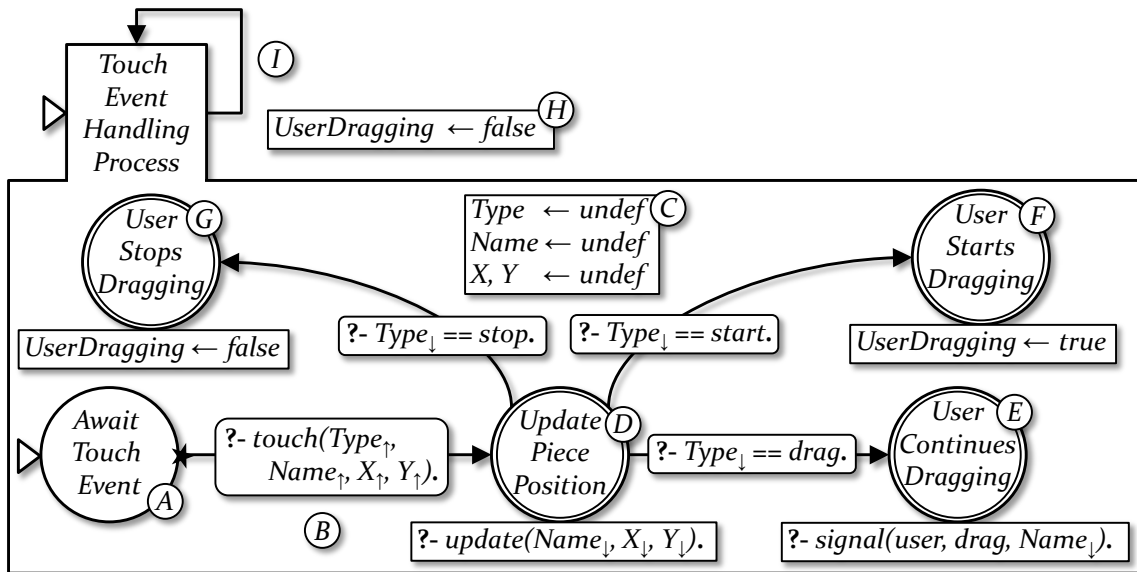


Figure 6.3.4: The behavior flow used for handling touch events received from the surface table.

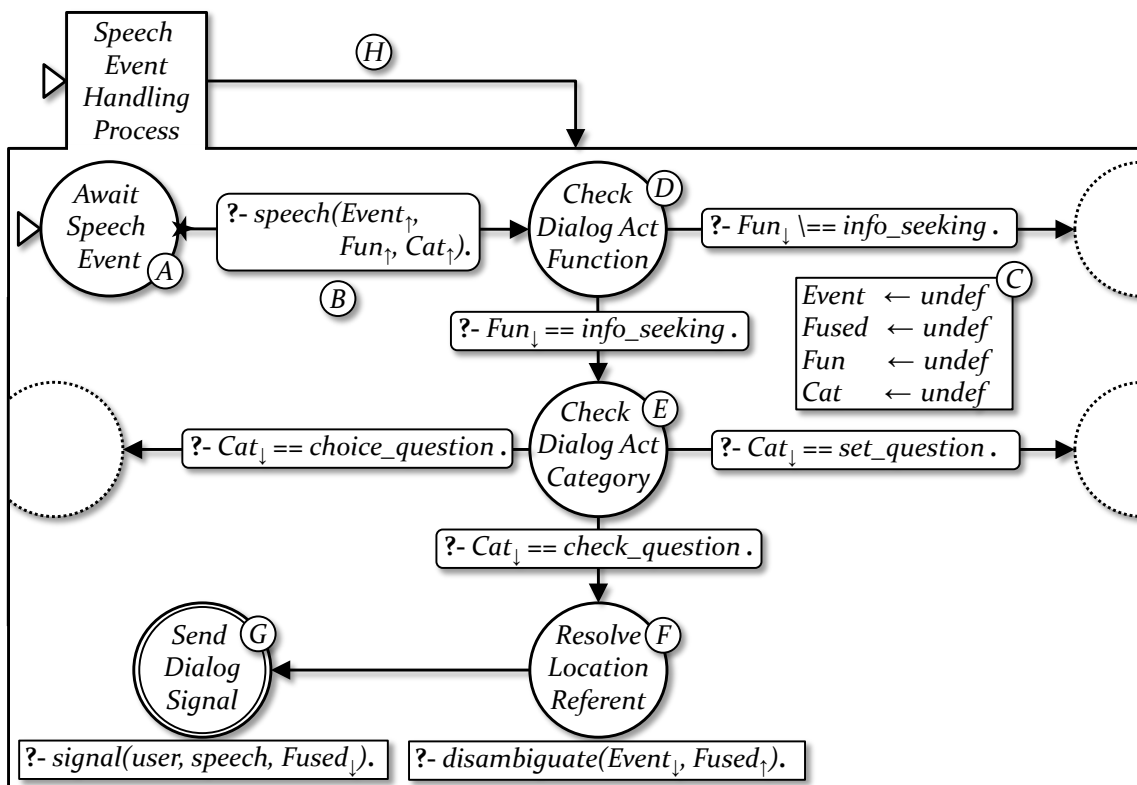


Figure 6.3.5: The behavior flow used for detecting and disambiguating the user's speech events.

Handling Speech Events

Figure 6.3.5 shows a simplified version of the behavior flow which is processing the user's *speech events* that are produced by the natural language understanding pipeline (Figure 6.1.2

ⓑ) by transcribing and semantically parsing the user's utterances into abstract dialog acts. The behavior flow starts by waiting for the user's next speech event to be asserted to the logic fact base (Figure 6.3.5 ⓐ). It uses the predicate *speech/3* (Figure 6.3.5 ⓑ) to extract a new speech event and store the event itself as well as the function and category of the contained dialog act in the local variables *Event*, *Fun* and *Cat* (Figure 6.3.5 ⓒ). Then, it checks if the dialog act is an *information seeking request* (Figure 6.3.5 ⓓ) and, if this is the case, examines whether the semantic category is a *check question* (Figure 6.3.5 ⓔ). In this case, as explained in Section 6.2.3, it tries to resolve potentially ambiguous verbal reference expressions with the user's gaze using the predicate *disambiguate/2* (Figure 6.3.5 ⓕ) and stores the eventually disambiguated speech event in the local variable *Fused* (Figure 6.3.5 ⓖ). A *speech action signal* is produced (Figure 6.3.5 ⓗ) to notify the behavior flow in Figure 6.3.24 that the multi-modal event has to be processed by the dialog management and propagated to the dialog planner as the user's next dialog contribution. Finally, the behavior flow restarts again to process the next incoming speech events of the user (Figure 6.3.5 ⓓ).

6.3.2 Behavioral Pattern Recognition

The *behavioral pattern recognition layer* (Figure 6.3.1 ⓔ) comprises behavior flows that continuously, concurrently, and incrementally recognize multi-modal and multi-directional behavioral patterns. Their decisions are based on changes of the global variables' values (Figure 6.3.1 ⓐ) and informed by the signals from the input event handling layer (Figure 6.3.1 ⓓ). Recognized patterns are signaled to the participant management (Figure 6.3.1 ⓕ) and the dialog and behavior control layers (Figure 6.3.1 ⓖ), thus, directly influencing the agent's floor and dialog flow management as well as the generation of role-specific ideomotor behaviors. The layer, for example, recognizes the coordinated use of touch actions, voice activities, and gaze shifts for the production of *turn-taking actions*, verbal and nonverbal *back-channels*, and *adjacency pairs*. Furthermore, *voice activity overlaps*, when the user barges into the agent's utterances or turn can be interpreted as *turn-taking conflicts*. Feedback inviting behaviors, such as *back-channel eliciting* and *facial mimicry eliciting* cues can also be detected. Finally, the participants' gaze movements are put into relation to monitor *gaze direction relations* and detect *gaze connection events*. In this, the layer exploits a major advantage of our modeling approach by recognizing each behavioral pattern in an individual parallel behavior flow. Integrating further patterns therefore solely requires to extend it with additional parallel behavior flows. Due to space and redundancy reasons, we only explain a selection of the most important and interesting behavior flows that can be realized on this layer.

Turn-Taking and Floor Management

Efficient collaboration requires the trouble-free regulation of the speaker right and the smooth exchange of the participant roles. This, in turn, requires the production and recognition of behavioral patterns commonly known as turn-taking or floor management actions (Traum and Rickel, 2002; Thórisson, 2002; Bohus and Horvitz, 2009, 2010c,a, 2011; Thórisson *et al.*, 2010). Our model orients along the *turn-taking actions* defined by Traum and Rickel (2002)

and the very similar *floor management actions* described by Bohus and Horvitz (2010c). These are well suited and largely acknowledged to describe multi-modal turn management in multi-party conversations. However, they do unfortunately not consider collaborative joint activities on shared workspaces. Those go beyond regular multi-modal dialogs since their participants can also produce turn-taking actions using application-specific contributions, such as moving an object on a table or pressing a button of a graphical user interface. In our case, the user may perform a turn, as usual, verbally, for example, by asking a question like those mentioned in Section 6.2.2, but may also produce very similar turn-taking actions by moving a puzzle piece to a position or puzzle field on the surface table. To cope with this notion of a turn, our model recognizes turn-taking actions when the user is only speaking or moving a puzzle piece on the surface table and even when he intermixes voice activities with object movements. A *turn take action* is then an attempt of the user to conquer the speaker or actor role by starting an utterance, moving an object or a combination of both. A *turn yield action* is produced when the user stops speaking or moving an object and is an attempt to offer the turn to one of the listeners but not necessarily the current addressee (Duncan, 1974; Argyle and Cook, 1976). A *turn assign action* is usually signaled by the speaker's directed gaze to a specific interaction partner at the end of the utterance or object movement and is used to explicitly select the next speaker or actor (Argyle and Cook, 1976). A *turn hold action* occurs at the boundaries of phrases or in short movement pauses and is an attempt to keep the turn at a point where one of the listeners or spectators might otherwise try to take the turn (Duncan, 1974; Argyle and Cook, 1976). We recognize a turn hold signal if the user continues speaking after a short speech pause in between two words or sentences and if he continues moving a puzzle piece after shortly holding the piece at a fix position for a moment.

Figure 6.3.6 shows the behavior flow that recognizes the user's *turn-taking actions* throughout a turn. It observes the user's voice and drag activity to decide if the user is claiming the turn at any point in time and produces *turn action signals* that are consumed by the higher layers of the model when the user *takes*, *holds*, *yields*, or *assigns* the turn. It is initially waiting for the user to start speaking or dragging an object (Figure 6.3.6 A) which is recognized when the global variables *UserSpeaking* and *UserDragging* (Figure 6.3.6 B) are set by the voice and touch event handling processes. If the user has started speaking or dragging an object, then it awaits some *take timeout* (Figure 6.3.6 C), defined by the local variable *Take* (Figure 6.3.6 D), before deciding that the user takes the turn. The short timeout avoids that accidental movements, verbal back-channels, ambient noise, or physiological reactions, like coughing or sneezing, are mistakenly interpreted as attempt to take the turn. If the user stops all voice and drag activity before the timeout has expired, then the behavior flow starts over again (Figure 6.3.6 A). Otherwise, a *turn take signal* is produced (Figure 6.3.6 E) and the global variable *UserClaiming* is updated (Figure 6.3.6 F) to notify the higher layers that the user has just taken and is now claiming the turn. Afterwards, the behavior flow waits until the user stops all voice and drag activity again (Figure 6.3.6 G) and thereupon waits for a certain *hold timeout* (Figure 6.3.6 H), defined by the local variable *Hold* (Figure 6.3.6 D). If this timeout expires, then the user might want to give up the turn and therefore a last *yield timeout* (Figure 6.3.6 I), defined by the local variable *Yield* (Figure 6.3.6 D), is awaited to decide if the

TURN
TAKING
ACTIONS

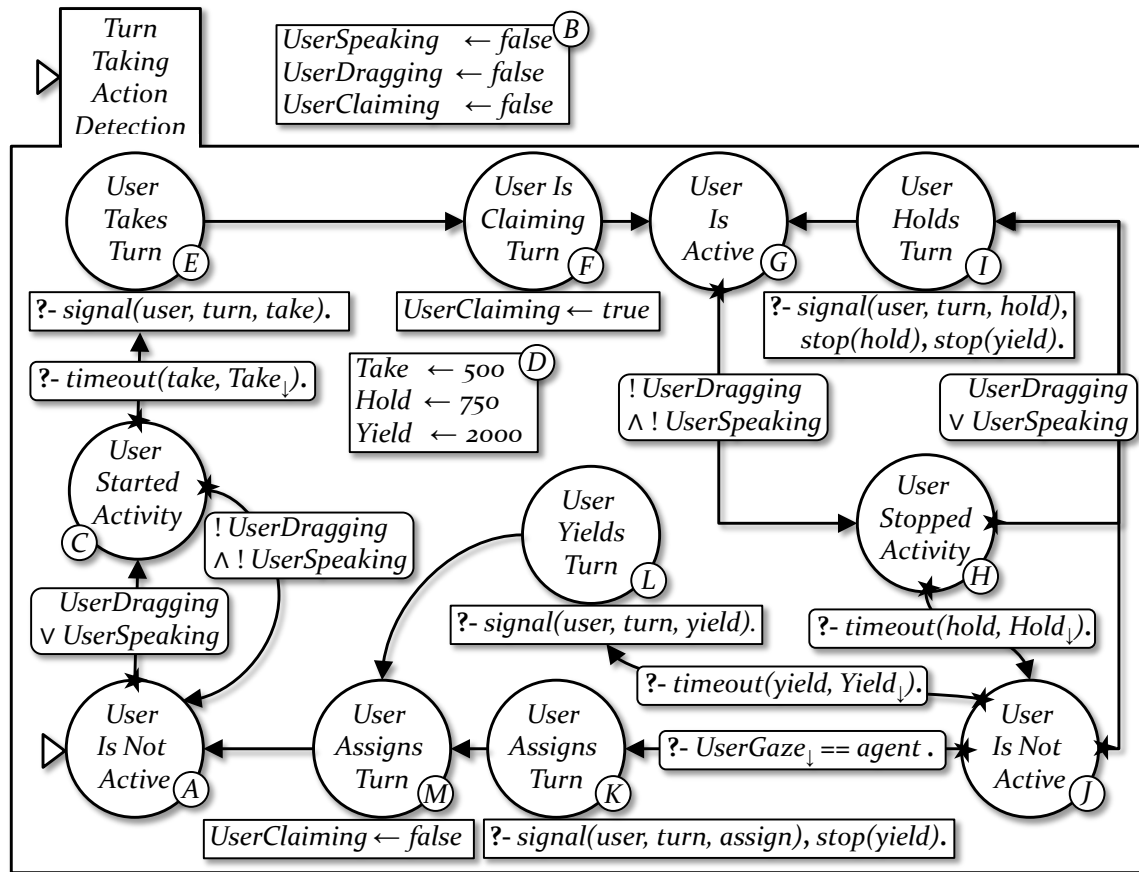


Figure 6.3.6: The behavior flow used for recognizing the user's different turn-taking actions.

user wants to directly assign the turn to the agent, just yield the turn or even hold the turn. If the user is looking at the agent, then this is interpreted as turn assign action and a *turn assign signal* is produced (Figure 6.3.6 (K)). The user's gaze target is retrieved via the global variable *UserGaze* that is set by the gaze event handling in Figure 6.3.3. Otherwise, if the yield timeout expires without the user looking at the agent, then a *turn yield signal* is produced (Figure 6.3.6 (L)). In both cases, the higher layers are informed that the user doesn't claim the turn anymore by resetting the global variable *UserClaiming* (Figure 6.3.6 (M)). During the hold or yield timeouts, the user might restart speaking or dragging an object to reclaim the turn before the timeouts have expired. In these cases, the behavior flow produces a *turn hold signal* (Figure 6.3.6 (I)) and starts waiting again (Figure 6.3.6 (G)). This gives the user the opportunity to correct himself in a second question or to overrule his questions in the turn by moving a puzzle piece. After passed through the turn, the behavior flow reenters the recognition process again (Figure 6.3.6 (A)). For choosing the timeout values we oriented along Rich *et al.* (2010) but also examined our own delays (Mehlmann *et al.*, 2014a,b).

Overlaps and Turn-Taking Conflicts

The recognition of the turn-taking actions described in Figure 6.3.6 is useful when modeling the basic sequential turn-taking model without overlaps and gaps (Sacks *et al.*, 1974). In such

a simplified serial model, the user's voice or action does never overlap with the agent's turn, such that there can never be recognized any interruption attempts when the user is barging into the agent's turn or vice versa. The agent is consequently not be able to interrupt itself when the user wants to conquer the speaker floor. However, highly interactive and natural interactions frequently reveal voice overlaps that could be interpreted as turn conflicts and cause interruptions. Thus, handling interruptions is a crucial aspect in the endeavor to give social agents human-like conversation skills (Ward *et al.*, 2015). The perception of interrupting and interrupted partner with respect to interpersonal attitudes, such as dominance, friendliness, closeness, politeness, and involvement can be influenced by changing the interruptibility as well as the strategy and timing of interrupt handling (Beattie, 1981a; Robinson and Reis, 1989; Murata, 1994; Tannen, 1994; Crown and Cummins, 1998; ter Maat *et al.*, 2010, 2011; Oviatt *et al.*, 2015; Cafaro *et al.*, 2016). For that reason, in addition to regular *turn-taking actions*, this layer also contains behavior flows detecting bidirectional *voice activity overlaps* and *turn-taking conflicts*. Figure 6.3.7 shows the hierarchical and parallel structure of the superordinate behavior flow which is divided into concurrently and closely synchronized, nested behavior flows. The first behavior flow is used for the retrospective classification of *voice overlap categories* (Figure 6.3.7 A) while the second one detects different *voice activity overlaps* and the third recognizes the resultant *voice overlap conflicts* in real-time (Figure 6.3.7 C). By analogy, the fourth behavior flow detects *voice-turn overlaps* (Figure 6.3.7 D) while the last one recognizes consequential *turn overlap conflicts* (Figure 6.3.7 E).

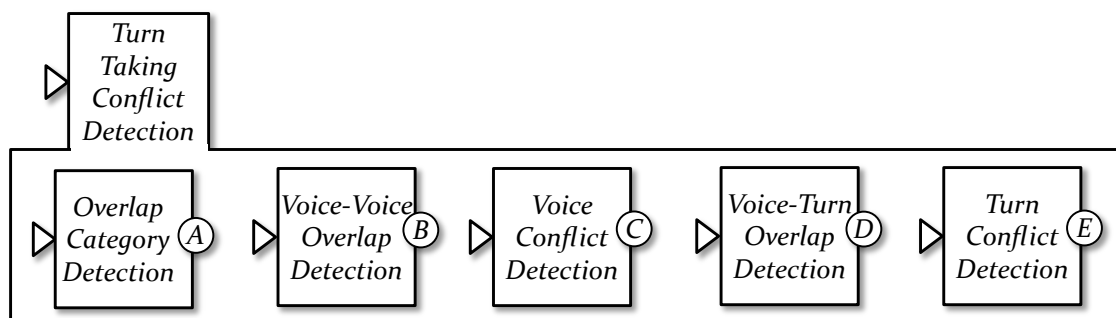


Figure 6.3.7: The behavior flow used for recognizing overlapping states and turn-taking conflicts.

Figure 6.3.8 shows the behavior flow that detects the onset of a *voice activity overlap* and thus a possible interruption attempt as fast as possible which is necessary to ensure an immediate reaction of the behavior and interaction model in real-time. This model is the basis to recognize *successful interruptions* (Roger *et al.*, 1988) and distinguish *simple interruptions* (Ferguson, 1977) from overlaps. All these events come along with an overlap time, in which both dialog partners speak simultaneously for a certain period of time (Drummond, 1989). Therefore, the behavior flow constantly monitors the user's and the agent's voice activities, detects as soon as their voices are overlapping and immediately notifies the higher layers of the model by updating corresponding global variables. The behavior flow starts by waiting until either the agent or the user starts speaking first (Figure 6.3.8 A) by observing the variables *UserSpeaking* and *AgentSpeaking* (Figure 6.3.8 B) that are set by the voice event

VOICE
ACTIVITY
OVERLAPS

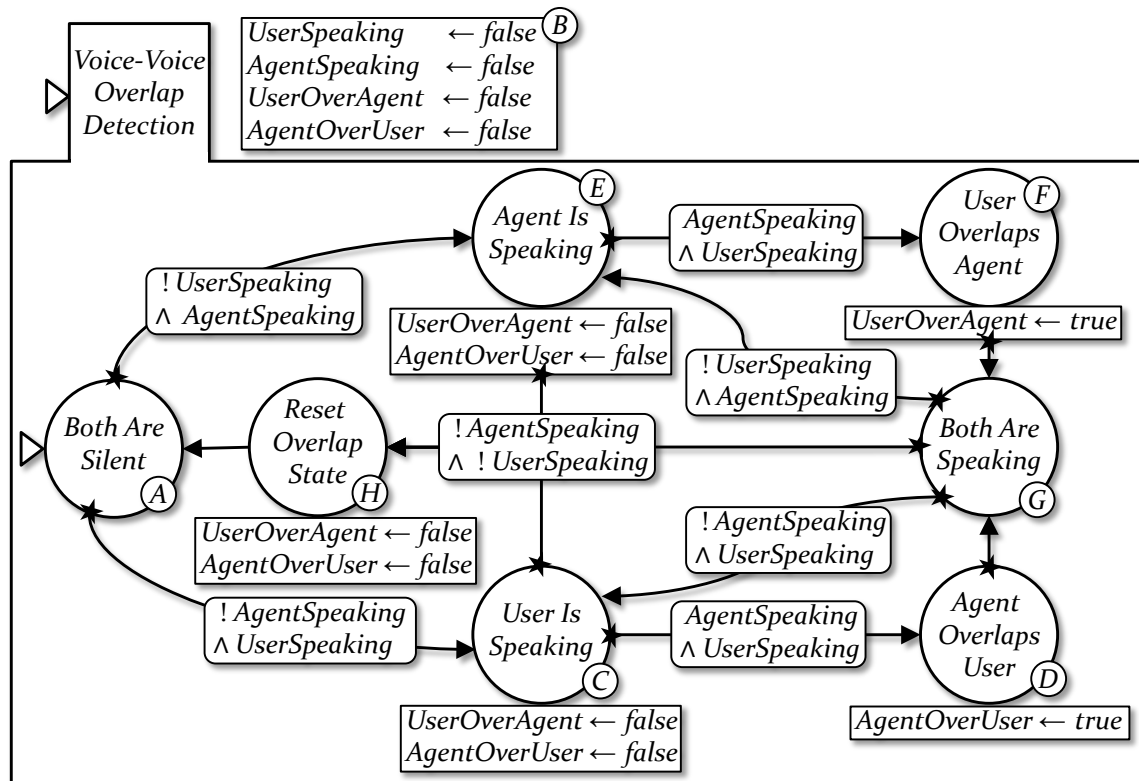


Figure 6.3.8: The behavior flow used for recognizing different directions of voice activity overlaps.

handling process. If the user starts first (Figure 6.3.8 ©), then it waits until either the user stops again (Figure 6.3.8 ⑧) and starts over (Figure 6.3.8 ①), or until the agent starts speaking too (Figure 6.3.8 ④), such that the agent’s voice overlaps the user’s speech (Figure 6.3.8 ④). While, the overlap detection the other way round works analogously (Figure 6.3.8 ⑤,⑥), in both cases, the global variables *UserOverAgent* or *AgentOverUser* (Figure 6.3.8 ②) are updated to inform the higher layers of the model which direction of voice activity overlap has just be detected. The overlap state is left when the agent (Figure 6.3.8 ④) or the user (Figure 6.3.8 ⑤) stops speaking. If both stop at the same time (Figure 6.3.8 ⑧), then it restarts to detect further voice overlaps (Figure 6.3.8 ①).

VOICE OVERLAP CONFLICTS The voice activity overlap model in Figure 6.3.8 is tightly synchronized with the behavior flow shown in Figure 6.3.9 which incrementally recognizes *voice overlap conflicts*, one of the possible *turn-taking conflicts*. It detects a turn-taking conflict as soon as a voice activity overlap lasts longer than a certain period of time. Such a conflict is then reported to the higher layers by producing appropriate *turn conflict signals* and updating corresponding global variables. The behavior flow starts by waiting until one of the two possible voice activity overlap states is entered (Figure 6.3.9 ①) which is recognized by a change of the variables *UserOverAgent* or *AgentOverUser* (Figure 6.3.9 ②). While the user overlaps the agent, it waits for a *barge-in timeout* (Figure 6.3.9 ④), defined by the local variable *BargeIn* (Figure 6.3.9 ③) and starts over again (Figure 6.3.9 ①) if the overlap stops before the timeout has expired. Otherwise, it produces a *turn barge-in signal* (Figure 6.3.9 ⑤), updates the global variable *UserBargingIn*

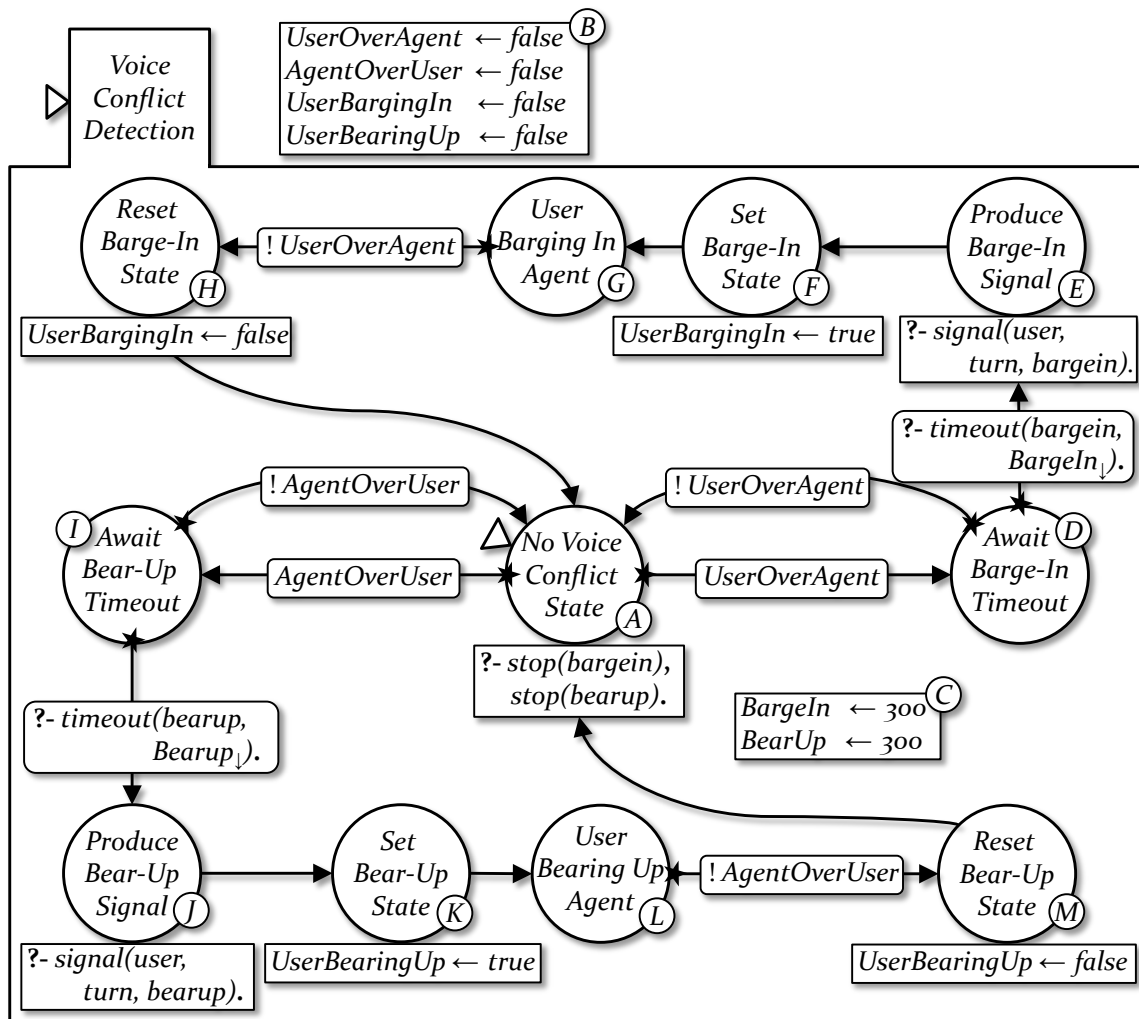


Figure 6.3.9: The behavior flow used for recognizing the different types of voice overlap conflicts.

(Figure 6.3.9 (F)) and remains in the barge-in state (Figure 6.3.9 (G)) until the voice overlap stops again which is then notified to the higher layers by resetting the global variable *UserBargingIn* (Figure 6.3.9 (H)) before starting over again (Figure 6.3.9 (A)). Finally, the overlap conflict detection the other way round works analogously (Figure 6.3.9 (I-M)).

It can often be observed that users barge in the agent's turn during the short speech pauses between two individual words or utterances of the agent's turn because they might believe that this is a possible completion point of the partner's turn (Sacks *et al.*, 1974). According to Li (2001) these situations would be categorized as interruptions without overlapping voice activity and are therefore also referred to as *silent interruptions* (Ferguson, 1977). The models in Figures 6.3.8 and 6.3.9 are not able to handle these situations as required since they are based exclusively on voice activity and not on whole turns. The model in Figure 6.3.8 would possibly not recognize any overlap while the model shown in Figure 6.3.9 might only recognize a bear-up conflict with an unwanted delay. However, fortunately they can easily be adapted to handle these cases by only changing a single variable and the name of the re-

TURN
OVERLAP
CONFLICTS

sulting conflict signals. Conflicts that cause silent interruptions because the user barges into the agents turn within a speech pause can easily be recognized by using a turn state variable *AgentClaiming* instead of the speaking state variable *AgentSpeaking*. This variable is true as long as the agent is claiming the turn, that means, also in short pause between two coherent words or utterances of the same turn. It is set by the agent's executor instance based on the notifications from the agent's text-to-speech engine. The corresponding behavior flows (Figure 6.3.7 ④, ⑤) do not need to distinguish barge-ins and bear-ups but simply create one and the same conflict signal in all cases.

Gaze Relations and Connections

Figure 6.3.10 shows the hierarchical and parallel structure of the behavior flow which is responsible for the detection of gaze relations and connections. A first nested behavior flow is recognizing the three *gaze direction relations* between the user's and the agent's gaze, namely *separated gaze*, *shared gaze*, and *mutual gaze* (Figure 6.3.10 ①). A second one is incrementally recognizing the two bidirectional *gaze connection events* defined by Rich *et al.* (2010) and Holroyd *et al.* (2011), namely *directed gaze* and *mutual facial gaze* (Figure 6.3.10 ②). To avoid confusions, it is important to mention that what Rich *et al.* (2010) refer to as *directed gaze* is correctly said *shared gaze* according to the standard terminology.

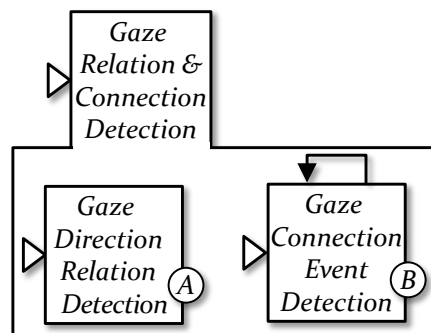


Figure 6.3.10: The behavior flow used for recognizing different gaze relations and connection events.

THE GAZE DIRECTION RELATIONS

Figure 6.3.11 shows the behavior flow that monitors the *gaze direction relations* between the two interaction partners. It constantly observes the user's and agent's gaze targets to determine if the user and the agent have established *shared gaze*, *mutual gaze* or neither of the two in order to keep the higher layers of the model updated about the current gaze direction relation. The behavior flow starts observing the global variables *UserGaze* and *AgentGaze* (Figure 6.3.11 ①) while user and agent do not share the same gaze target. A nested behavior flow is constantly checking if the two participants are establishing mutual gaze (Figure 6.3.11 ②). It remains in the initial state while both are neither looking at each other nor sharing the same gaze target (Figure 6.3.11 ③). When they look at each other, then it enters the mutual gaze state (Figure 6.3.11 ④) and updates the global variable *MutualGaze* (Figure 6.3.11 ⑤). It is immediately interrupted when the user and the agent share the same gaze target such that the superordinate behavior flow updates the variables *SharedGaze* and *MutualGaze* (Fig-

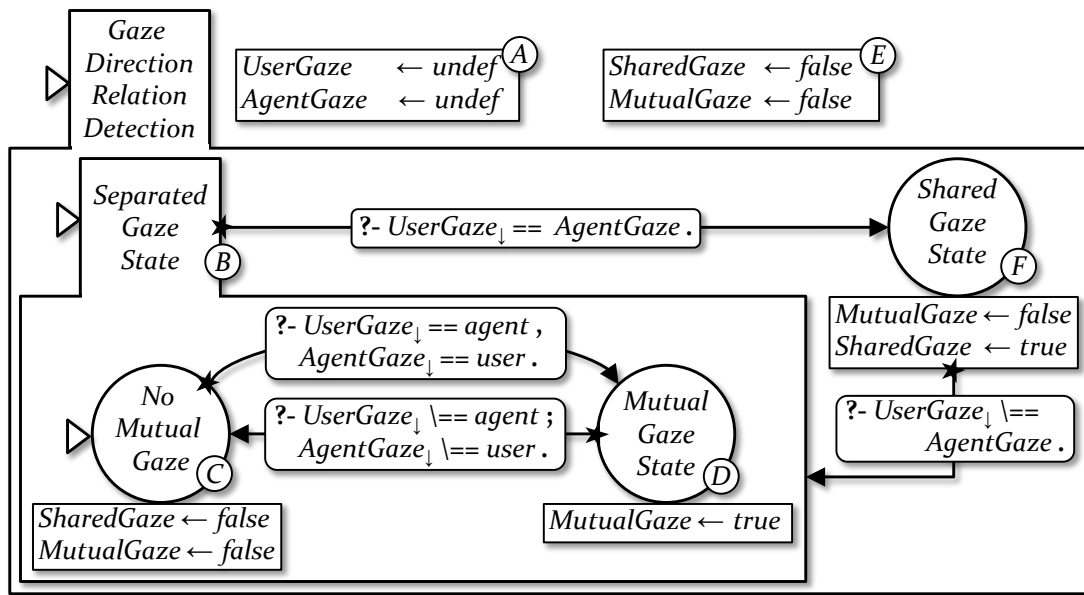


Figure 6.3.11: The behavior flow used for monitoring gaze relations between the user and the agent.

ure 6.3.11 (E) and remains in the shared gaze state (Figure 6.3.11 (F)) until they look at different gaze targets again (Figure 6.3.11 (B)).

Figure 6.3.12 shows the behavior flow which is used to recognize the *gaze connection events* as defined by Rich *et al.* (2010) and Holroyd *et al.* (2011) (Figure 6.3.10 (B)). The behavior flow recognizes whenever the participants directly follow each other's gaze shifts with the intention to establish shared gaze, immediately answer each other's attempts to establish mutual facial gaze, or whether they focus their visual attention to different targets. When one of the participants focuses his attention on an object or the partner, then the other has the possibility to react within a specific time window to successfully produce a gaze connection event. If the other partner chooses a different gaze target or doesn't even respond before the timeout has expired, then the partners fail to create this gaze connection event.

THE GAZE
CONNECTION
EVENTS

The behavior flow starts by waiting for the next gaze shift of one of the two participants which is signaled by a *gaze shift signal* (Figure 6.3.12 (A)) produced by the gaze event handling process. When one of the participants shifts his gaze (Figure 6.3.12 (B)), then the partner's next gaze shift is awaited (Figure 6.3.12 (B)) during a *connect timeout*, defined by the local variable *Connect* (Figure 6.3.12 (C)). If no gaze shift happens before the timeout expires (Figure 6.3.12 (E)), then the connection event detection restarts again (Figure 6.3.12 (F)). Otherwise, the two partners' last gaze targets are compared (Figure 6.3.12 (D)). If they look at each other then a *mutual facial gaze signal* is produced (Figure 6.3.12 (G)), if both focus on the same gaze target, then a *directed gaze signal* is produced (Figure 6.3.12 (H)). In any case, even if no connection event is detected, the behavior flow starts over again (Figure 6.3.12 (F)).

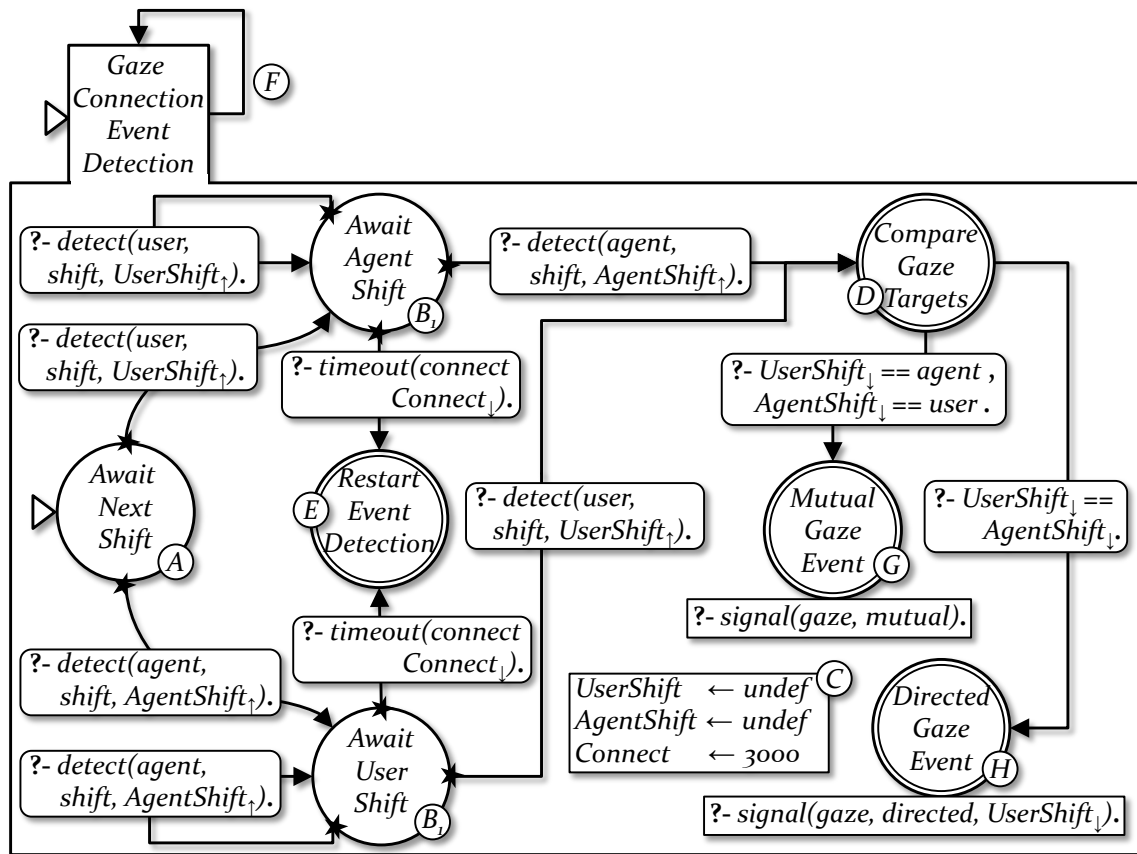


Figure 6.3.12: The behavior flow used for recognizing mutual facial gaze and directed gaze events.

Feedback and Response Behaviors

Figure 6.3.13 shows the hierarchical and parallel structure of the behavior flow which is responsible for the recognition of different *feedback behaviors* which are characterized by the close temporal alignment and meshing of both participant’s verbal and nonverbal behaviors. A first nested behavior flow is used for the detection of *verbal back-channels* (Figure 6.3.13 (A)), a second is responsible for the recognition of *nonverbal back-channels* (Figure 6.3.13 (B)) and a third is used for the detection of *adjacency pairs* or *direct responses* (Figure 6.3.13 (C)).

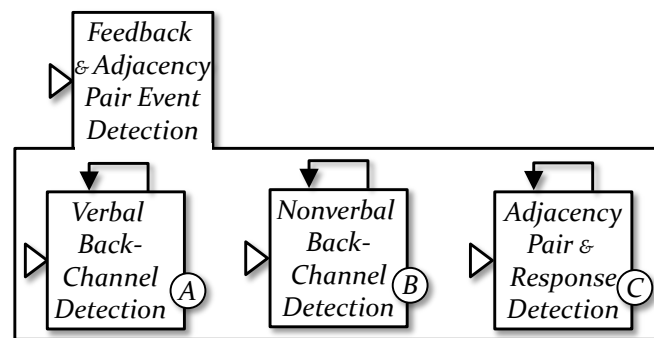


Figure 6.3.13: The behavior flow used for recognizing back-channels, adjacency pairs and responses.

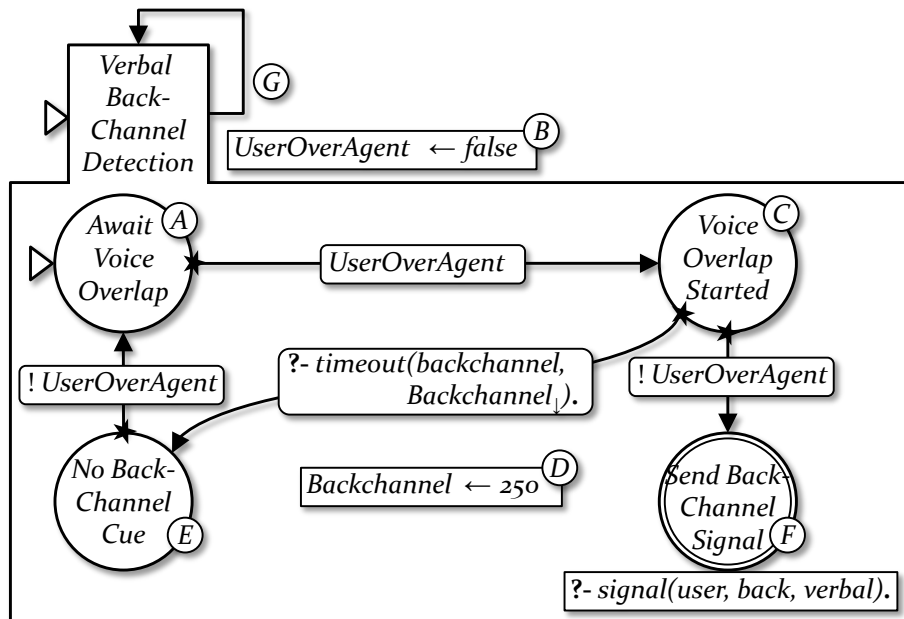


Figure 6.3.14: The behavior flow used for recognizing the user’s verbal back-channel statements.

Figure 6.3.14 shows the behavior flow that recognizes *verbal back-channels* (Yngve, 1970; Lambertz, 2011), which are a listener’s short verbal statements like “hmm” or “ahh” while listening to the speaker but without claiming the turn. A verbal back-channel cue is recognized, independent of the participant role division, when the user’s and the agent’s voices are overlapping for shorter than a predetermined period of time, usually without raising a turn-taking conflict. The behavior flow starts by waiting for the user to overlap with the agent’s speech (Figure 6.3.14 A) by observing the global variable *UserOverAgent* (Figure 6.3.14 B) which is set by the overlap detection process. During the overlapping state (Figure 6.3.14 C), it waits for a specific *back-channel timeout*, defined by the local variable *Backchannel* (Figure 6.3.14 D), until it assumes that the overlap is too long to be considered as back-channel (Figure 6.3.14 E) and thereupon waits until the overlap has ended again (Figure 6.3.14 A). If the overlap stops before the timeout has expired, then it is interpreted as a verbal back-channel statement and a *verbal back-channel signal* is produced (Figure 6.3.14 F) in order to be processed on the higher layers of the model. Afterwards, the behavior flow starts over again to detect consecutive back-channels (Figure 6.3.14 G).

DETECTING
VERBAL
BACK-CHANNELS

Figure 6.3.15 shows the behavior flow which is responsible for recognizing *nonverbal back-channel* behaviors (Kendon, 1967; Yngve, 1970; Allwood *et al.*, 1993; Bavelas *et al.*, 2002). A nonverbal back-channel cue is recognized, independent of the participant role division, whenever the user’s is performing a head nod or shake while the agent is speaking. The model can be further refined to cover additional nonverbal behaviors, such as facial expressions, like raising the eyebrows, or body gestures, such as shrugging of the shoulders. It can also be adapted such that it requires the user to look at the agent or to be in the addressee role when performing a back-channel cue. The behavior flow starts by waiting for the agent to

DETECTING
NONVERBAL
BACK-CHANNELS

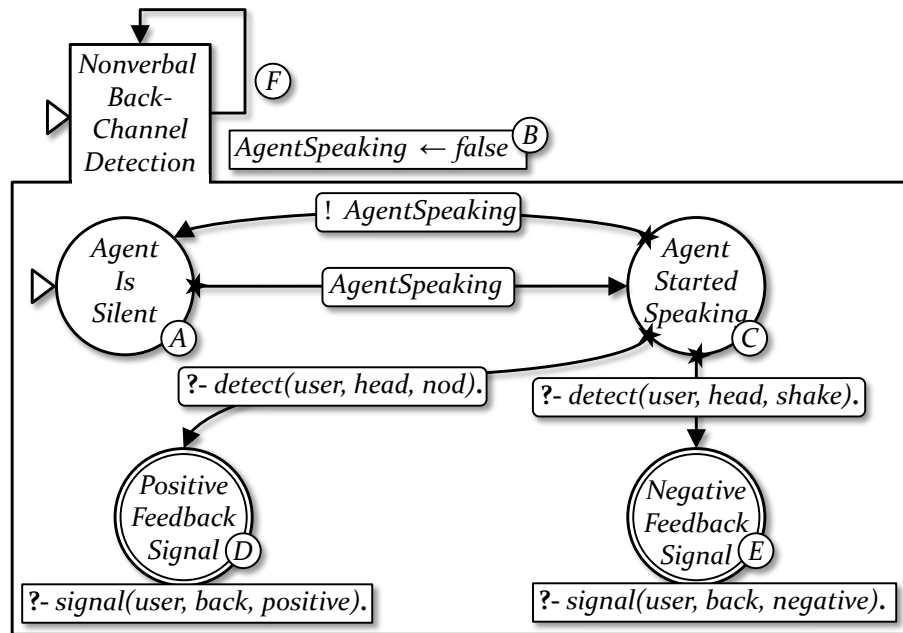


Figure 6.3.15: The behavior flow used for recognizing the user’s nonverbal back-channel behaviors.

start speaking (Figure 6.3.15 (A)) by observing the global variable *AgentSpeaking* (Figure 6.3.15 (B)) which is set by the voice event handling process. After the agent has started speaking (Figure 6.3.15 (C)), it waits for a *head movement signal* produced by the input event handling layer. A *head nod signal* is interpreted as confirmation, acceptance or positive back-channel cue, in general, and the behavior flow consequently produces a *positive back-channel signal* (Figure 6.3.15 (D)). A *head shake signal* is considered as expression of dislike or rejection and causes the creation of a *negative back-channel signal* (Figure 6.3.15 (E)). These back-channel signals can then be further processed by the behavior flows on the higher layer of the model. Finally, the behavior flow restarts to detect the next back-channel cues (Figure 6.3.15 (F)).

DETECTING ADJACENCY PAIR EVENTS

Figure 6.3.16 shows the behavior flow which is used for the detection of *adjacency pair events* as described by Rich *et al.* (2010) and Holroyd *et al.* (2011) which are basically the same as the *direct response events* defined by Bohus and Horvitz (2011). These bidirectional patterns consist of a pair of utterances where the first one provokes the second, responding utterance within a particular period of time. Just like back-channel cues, the frequency of recognized adjacency pairs is considered as a measure for the engagement between the two interaction partners (Rich *et al.*, 2010; Holroyd *et al.*, 2011). The behavior flow starts by waiting that either the user or the agent starts speaking (Figure 6.3.16 (A)) by observing the global variables *UserSpeaking* and *AgentSpeaking* (Figure 6.3.16 (B)). If the user starts speaking first (Figure 6.3.16 (C)), then the behavior flow waits until the user stops speaking again while the agent remains silent (Figure 6.3.16 (D)). In this case, the behavior flow waits for a response of the agent until a specific *response timeout*, defined by the local variable *Response* (Figure 6.3.16 (E)), has expired. Vice versa, if the agent starts speaking first (Figure 6.3.16 (F)) and then stops again while the user keeps silent (Figure 6.3.16 (G)), then the user’s response is awaited during the

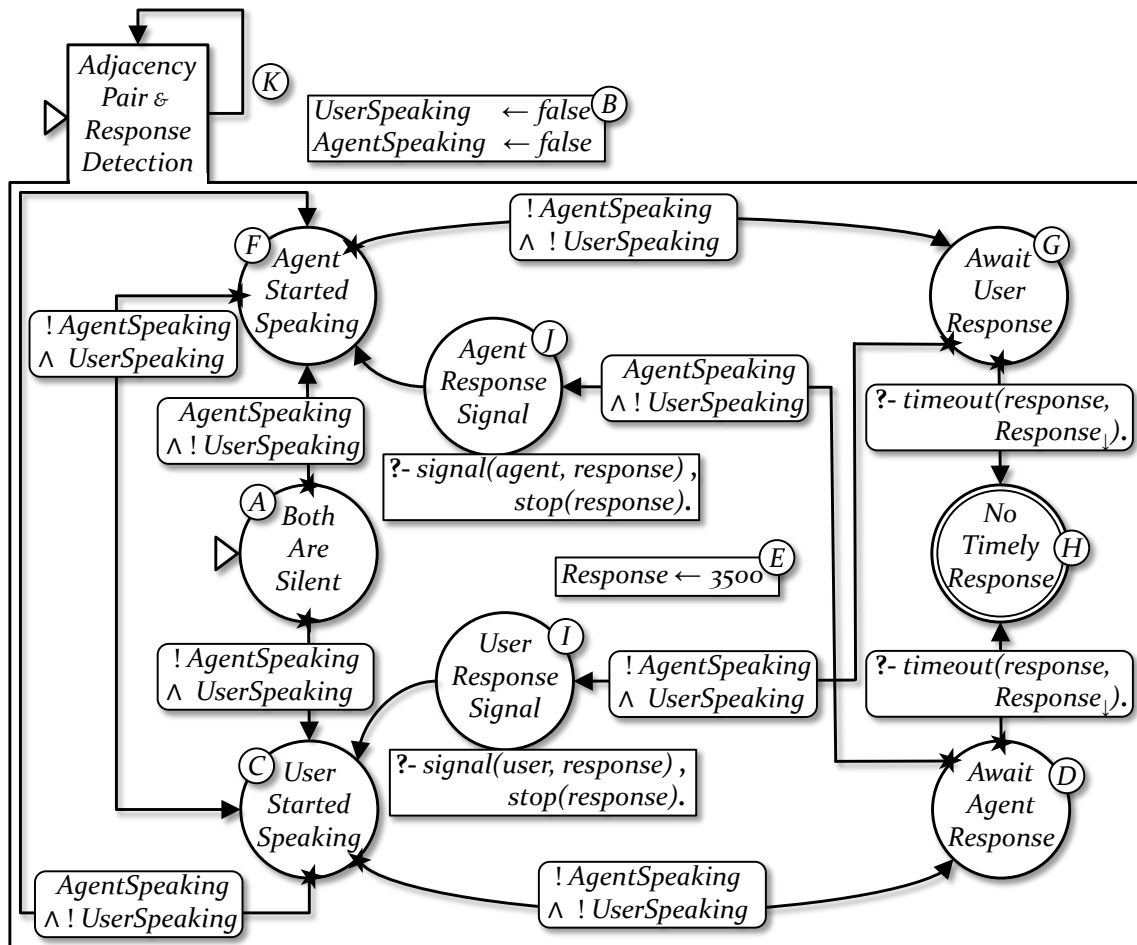


Figure 6.3.16: The behavior flow used for recognizing adjacency pair events or direct response events.

timeout. If the participants' responses happen before the timeout has elapsed, then a *user response signal* (Figure 6.3.16 (I)) or *agent response signal* (Figure 6.3.16 (I)), respectively, is produced and is further processed by the higher layers of the model. Otherwise, if the silence gap takes longer (Figure 6.3.16 (H)), then the behavior flow restarts (Figure 6.3.16 (K)).

Back-Channel and Mimicry Eliciting

Figure 6.3.17 shows the hierarchical and parallel structure of the behavior flow that is responsible for the recognition of different *feedback eliciting cues*. A first nested behavior flow is used for the detection of *back-channel eliciting cues* (Figure 6.3.17 (A)) while a second is responsible for the recognition of *facial mimicry eliciting behaviors* (Figure 6.3.17 (B)).

Figure 6.3.18 shows the behavior flow which is recognizing *back-channel eliciting cues* as observed by Kendon (1967) and Bavelas *et al.* (2002) and studied by Oertel *et al.* (2012) and Hjalmarsson and Oertel (2012). In line with their findings, in our model, a back-channel inviting cue is recognized if the user is performing a glance of facial gaze to the agent while he is speaking or moving a puzzle piece on the surface. The behavior flow starts with waiting for the user to claim the turn (Figure 6.3.18 (A)) by observing the global variable *UserClaiming*

BACK-CHANNEL
ELICITING
BEHAVIORS

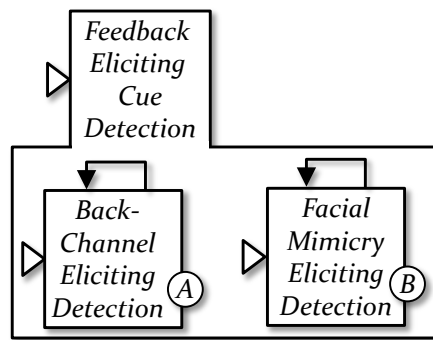


Figure 6.3.17: The behavior flow used for recognizing the different types of feedback eliciting cues.

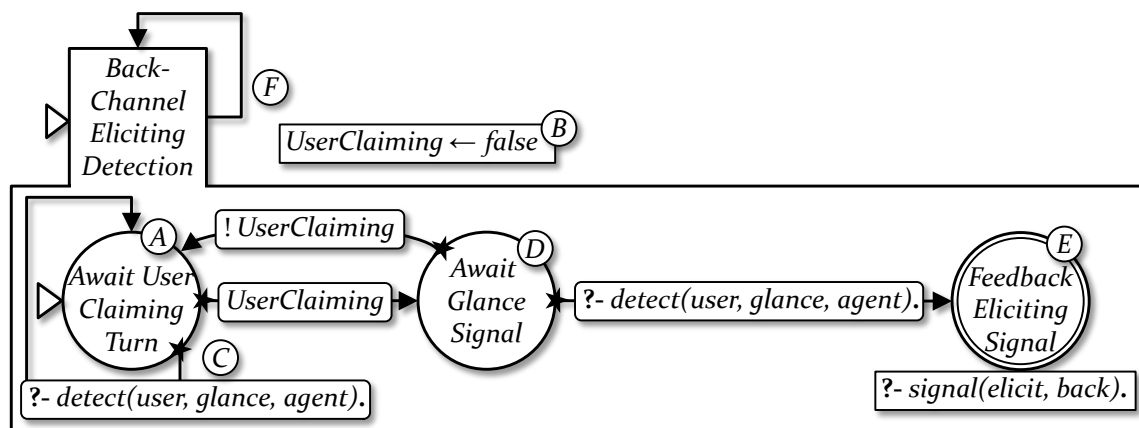


Figure 6.3.18: The behavior flow interpreting the user's gaze glances as back-channel eliciting cues.

(Figure 6.3.18 (B)) which is set by the turn-taking action recognition process. In this, it ignores all attempts of the user to establish mutual facial gaze (Figure 6.3.18 (C)). When the user is claiming the turn, then it waits for a *gaze glance signal* (Figure 6.3.18 (D)) from the gaze event handling process which is then interpreted as back-channel inviting cue and causes a *back-channel eliciting signal* (Figure 6.3.18 (E)) that is directed to the behavior flow which controls the agent's ideomotor nonverbal behavior. Afterwards, the behavior flow starts over again to detect eventually following eliciting cues (Figure 6.3.18 (F)).

FACIAL MIMICRY ELICITING

Figure 6.3.19 shows the behavior flow which can detect *facial mimicry eliciting* behaviors. They can cause catching the other's emotion when being looked at with an emotional expression (Hess and Fischer, 2013; Chartrand and Lakin, 2013) no matter if this mirroring behavior represents an unconscious emotional contagion or an intentional mimicry or imitation of the partner's behavior (Louwerse et al., 2012; Chartrand and Lakin, 2013). In our model, the agent decides how long and intense to mimic the user's facial expressions whenever the user tries to establish mutual facial gaze while claiming the turn and showing an emotional display. The behavior flow starts with waiting for the user to claim the turn (Figure 6.3.19 (A)) by monitoring the global variable *UserClaiming* (Figure 6.3.19 (B)) which is changed by the turn-taking action recognition process. In this, it ignores the user's emotion displays (Fig-

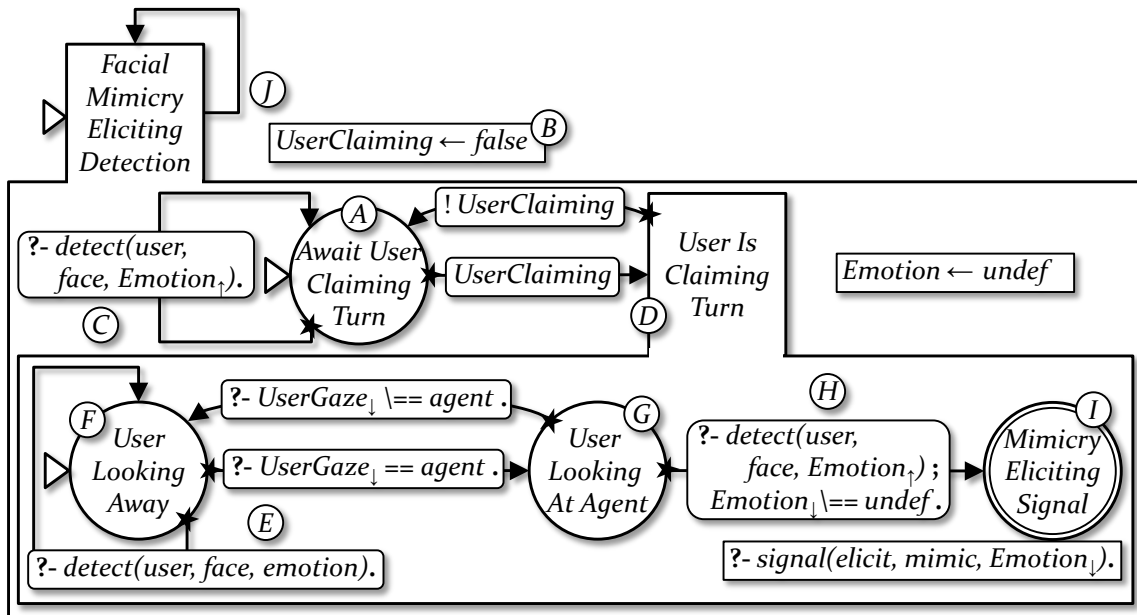


Figure 6.3.19: The behavior flow which is used for recognizing the user’s facial mimicry eliciting cues.

ure 6.3.19 ©) until the user is claiming the turn (Figure 6.3.19 ④). Then, it still ignores the user’s emotions (Figure 6.3.19 ⑤) while the user is not yet looking at the agent (Figure 6.3.19 ⑥). When the user looks at the agent (Figure 6.3.19 ⑦), then it reacts to an already detected or the next recognized facial expression (Figure 6.3.19 ⑧) by producing a *facial mimicry eliciting signal* (Figure 6.3.19 ⑨) which is then consumed by the behavior flow in that controls the agent’s nonverbal behavior on a higher layer of the behavior and interaction model. Finally, the behavior flow starts over again to detect the next eliciting cues (Figure 6.3.18 ⑩).

6.3.3 Participant Role Management

The behavior flows on the *participant role management layer* (Figure 6.3.1 ⑥) are responsible for the assignment of the different participant roles (Goffman, 1979; Clark and Carlson, 1982; Wilkes-Gibbs and Clark, 1992; Clark, 1996) to the interaction partners and for coming to the decisions when to shift these roles between them. These decisions depend, on the one hand, on the recently recognized turn-taking actions of the user and possible turn-taking conflicts, and, on the other hand, on the existence, urgency and importance of the current contribution that the agent is eventually willing to perform. Therefore, it exchanges information with the dialog and behavior control layer (Figure 6.3.1 ⑦) using *dialog flow signals* and is informed by the behavioral pattern recognition layer (Figure 6.3.1 ⑥) via the various types of *turn regulation signals*. The interruption policy implemented on the participant role management layer directly influences the dialog flow and the generation of the agent’s automatic, role-dependent, nonverbal behavior on the dialog and behavior control layer (Figure 6.3.1 ⑦).

The model uses roles that adapt the *overhearer*, *bystander*, *addressee*, and *speaker* roles used by Lee and Marsella (2011) and Bohus and Horvitz (2010a). An overhearer is not participating

in an interaction while the speaker and addressee are the core participants of the conversation. According to [Bohus and Horvitz \(2010a\)](#), bystanders are openly present in the environment but do not participate in the conversation. We adapted this role in our model, such that the agent is bystander whenever the speaker floor is not yet demanded or negotiated. This is the case at the very beginning of the interaction or when the floor has been offered by a partner but has not been accepted by the other within a certain period of time. We also extended the speaker role to a general actor role, such that the user is also in the speaker role when he has successfully taken the turn with an object dragging action. In addition, we introduce two transition states between these roles, first the *offering phase* which is entered after the agent has assigned the turn to the user and the *offered phase* which is entered after the user assigns the turn to the agent. Both transition phases end after some timeout if the addressee does not accept the offer by taking the turn. In this case both partners enter the bystander role until one of them wants to contribute to the interaction again. For the timing in shifting roles via these transition states we use a time window similar to those proposed by [ter Maat et al. \(2011\)](#) and [Smith et al. \(2015\)](#). These extensions allow more flexibility and refinement in designing role dependent automatic behavior than with only two strict core roles. For example, while the agent could actively follow the user's attention to puzzle pieces in the addressee role, it could try to establish mutual gaze in the offering phase and only show some random behavior in the bystander role. They also offer more flexibility to model the interruption policy for the collaborative shared workspace scenario. For example, as dragging an object is not directly addressed to the agent, we could decide to ignore the user's turn take action, which could possibly result from such a dragging action, and only interrupt the agent if there can be observed an actual voice overlap conflict.

Realizing Participant Role Shifts

Figure 6.3.20 shows an extract of the behavior flow that manages the participant role assignment and shifting. It is hierarchically refined into two nested behavior flows that are executed depending on the value of the global variables *UserPresent* and *AgentPresent* (Figure 6.3.20 ①) which are set on the input handling layer (Figure 6.3.1 ①). The execution remains in the first behavior flow (Figure 6.3.20 ②) as long as one of the interaction partners has not yet joined the interaction. In this case, the *overhearer* role is assigned to the agent by setting the global variable *AgentRole* (Figure 6.3.20 ③). When both, user and agent are taking part in the interaction, then the second behavior flow is controlling the shifting of the other participant roles (Figure 6.3.20 ④). These role shifts are induced by *floor taking signals* produced by the nested behavior flows (Figure 6.3.20 ⑤-⑥) which, for their part, produce these signals in reaction to *turn regulation signals* of the behavioral pattern recognition layer and *dialog flow signals* from in the dialog and behavior control layer. When assigning roles, then the behavior flow updates the global variables *AgentRole* and *RoleAction* (Figure 6.3.20 ③) representing the agent's role and action, respectively. The nested behavior flow (Figure 6.3.20 ④) starts in the state in which both participants are in the bystander role (Figure 6.3.20 ⑤) until either the agent or the user occupies the speaker role caused by a *floor occupy signal*. The agent occupies the speaker floor when successfully requesting a new dialog contribution

whereas the user occupies the floor when he takes the turn. Depending which of them occupies the floor first, the behavior flow either assigns the speaker role to the user and the addressee role to the agent (Figure 6.3.20 ⑥) or vice versa (Figure 6.3.20 ⑦).

While the agent is the addressee (Figure 6.3.20 ⑥), the corresponding nested behavior flow produces a *floor conquer signal* if the agent requests a dialog contribution while the user is interruptible, such that the agent immediately takes on the speaker role (Figure 6.3.20 ⑧). If the user holds the turn, then a *floor claim signal* causes that participant roles remain unchanged (Figure 6.3.20 ⑥) and if the user yields the turn, then a *floor release signal* causes that both participants return to the bystander role (Figure 6.3.20 ⑤). If the user assigns the turn and thus offers the speaker floor to the agent, then a *floor offer signal* causes that the behavior flow enters the corresponding transition state (Figure 6.3.20 ⑨). Similar to the strategy described by ter Maat and Heylen (2009); ter Maat *et al.* (2010) and ter Maat *et al.* (2011), the agent then has some time window to make sure that the user really finished his contribution and to accept the offered role exchange by starting a contribution. During this *transition timeout*, defined by the local variable *Transition* (Figure 6.3.20 ⑩), the user can take the turn again to produce a *floor reclaim signal* and thus withdraw his offer (Figure 6.3.20 ⑥). On the other hand, the agent can achieve the creation of a *floor accept signal* by requesting a dialog contribution before, such that the speaker role is assigned to him (Figure 6.3.20 ⑧).

While the agent is the speaker (Figure 6.3.20 ⑧), the user's turn take or hold actions as well as turn conflicts while the agent is interruptible can be transformed to a *dialog interrupted signal* and a *floor conquer signal* to make sure that the agent immediately interrupts itself and the user takes the speaker role (Figure 6.3.20 ⑥). Otherwise, if the agent's contribution is regularly finished, then a *floor offer signal* is created to offer the speaker floor to the user (Figure 6.3.20 ⑩). The user can then take the turn before the *transition timeout* has expired in order to produce a *floor accept signal* and thus accept the agent's offer (Figure 6.3.20 ⑬). However, the agent can also request an additional contribution before and thus effect the production of a *floor reclaim signal* to withdraw his offer (Figure 6.3.20 ⑧). If the user doesn't accept the offer in time and the agent does not reclaim the speaker role, then the bystander role is assigned to both and the behavior flow starts over again (Figure 6.3.20 ⑤).

As shown in Figure 6.3.22, if the user takes the turn and thus produces a *floor occupy signal* while the agent is in the speaker role and the agent subsequently offers the speaker role to the user before a turn conflict occurs, then neither a *dialog interrupted signal* nor a *floor conquer signal* are produced. However, the aforementioned *floor occupy signal* causes that the agent immediately switches to the addressee role after finishing its contribution and offering the floor to the user (Figure 6.3.20 ⑬). In turn, if the user first occupies and then immediately yields or assigns the floor to the agent while the agent still has the speaker role, then a *floor release signal* overwrites and thus cancels the *floor occupy signal* so that the user can again choose to accept the offer or not (Figure 6.3.20 ⑫). As shown in Figure 6.3.22, these situations usually occur when the agent's current contribution is not interruptible such that the agent first finishes his contribution before offering the speaker role to the user.

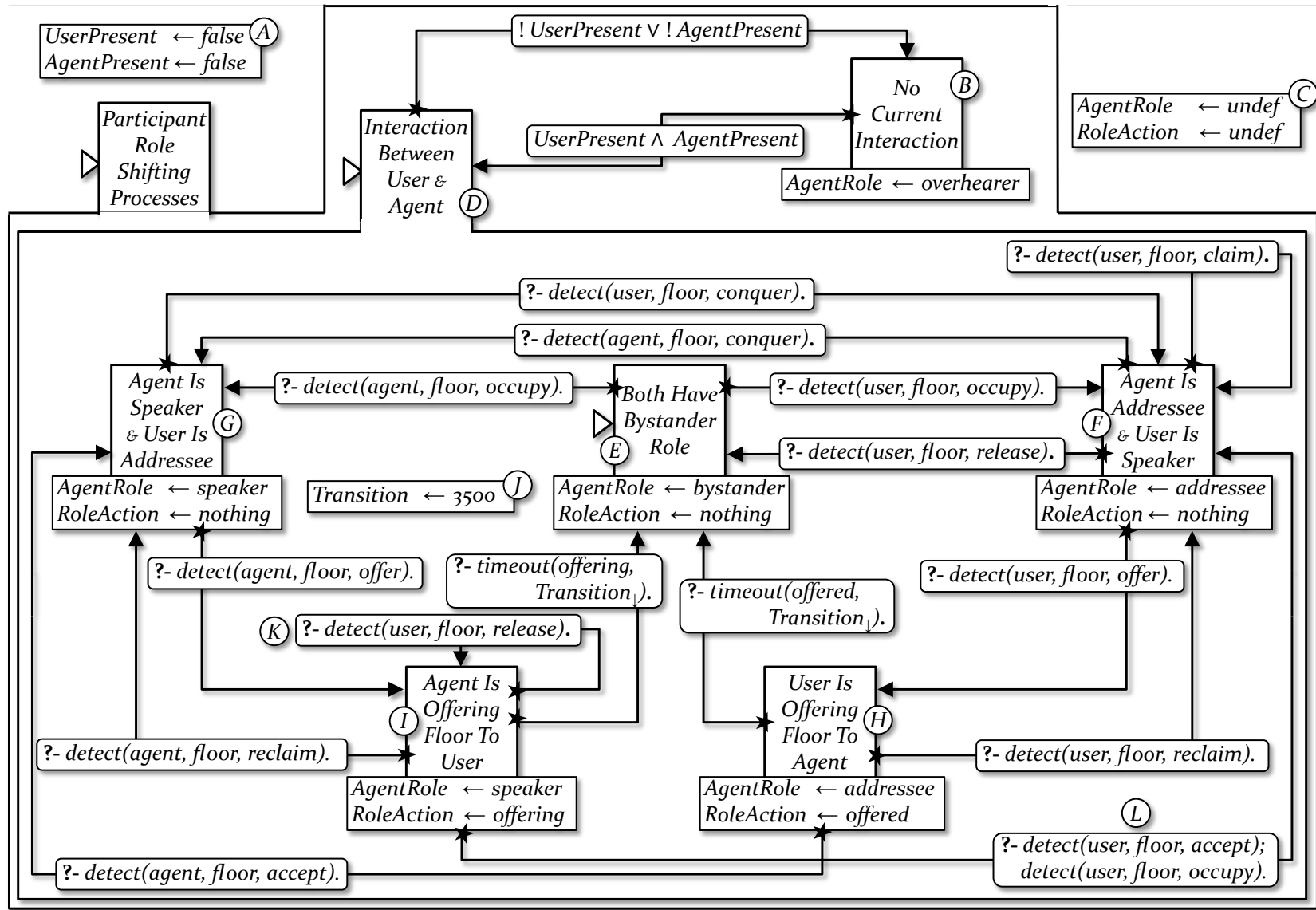


Figure 6.3.20: The behavior flow managing the participant role assignment and shifting based on turn regulation signals from other layers of the model.

Implementing Interruption Policies

The nested behavior flows are role-dependently transforming turn-taking actions, turn-taking conflicts, and dialog contribution requests into possible role shifts. In particular, they implement *interruption policies*, that are used to decide when such an event results in an interruption attempt. While the agent is the addressee (Figure 6.3.20 (F)), he might nevertheless want to contribute to the dialog because the dialog planner has produced a new contribution in the meanwhile. In this case, the *user interruption policy* has to decide whether the agent's contribution request may be accepted such that he may try to interrupt the user or not. Otherwise, while the agent is the speaker (Figure 6.3.20 (G)), the user could start to speak or drag an object and thus produce a turn take action or even a turn-taking conflict. In these cases, the *agent interruption policy* has to come to the decision if the agent has to interrupt itself and leave the speaker right to the user or not. While the model in the current form defines these interruption policies rather statically, they could just as well be based on politeness considerations (Brown and Levinson, 1987), interpersonal relationships and personality (Zimmerman and West, 1975; Ferguson, 1977; Natale et al., 1979; Goldberg, 1990), the urgency and importance of individual the agent's contributions, and many other factors that determine under what circumstances one participant may interrupt the other.

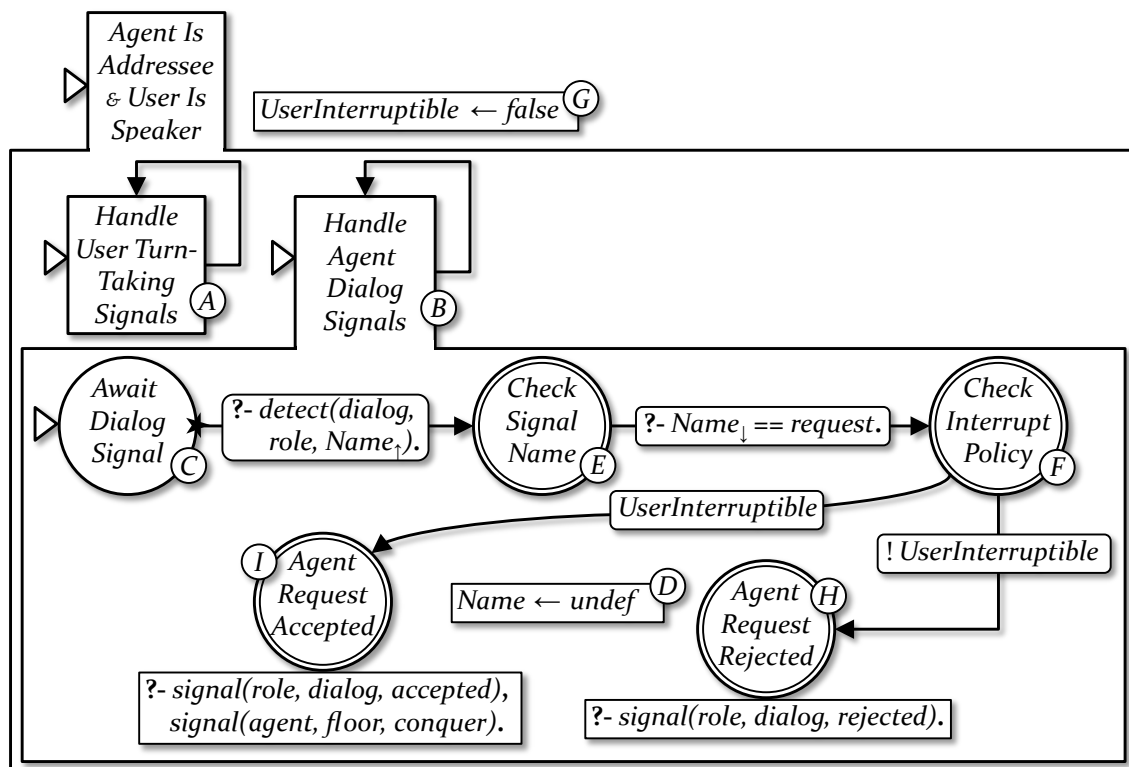


Figure 6.3.21: The behavior flow used for handling turn-taking and dialog signals as addressee.

Figure 6.3.21 shows the behavior flow which is used to decide if a role shift has to be performed while the agent is in the addressee role (Figure 6.3.20 (F)). A first nested behavior flow is reacting to the user's turn-taking actions (Figure 6.3.21 (A)) while the second behavior

THE USER
INTERRUPTION
POLICY

flow (Figure 6.3.21 (B)) is implementing the interruption policy based on *dialog flow signals*. It waits until it detects such a signal in the fact base (Figure 6.3.21 (C)), extracts its name to the local variable *Name* (Figure 6.3.21 (D)) and afterwards checks which action may have caused the signal (Figure 6.3.21 (E)). In the case of a *floor request signal*, it checks if the user is interruptible (Figure 6.3.21 (F)) by checking the corresponding global variable *UserInterruptible* (Figure 6.3.21 (G)). If the user is not interruptible, then the behavior flow produces a *dialog reject signal* (Figure 6.3.21 (H)) to ensure that the contribution is refused. Otherwise, it first produces an *dialog accept signal* and afterwards a *floor conquer signal* (Figure 6.3.21 (I)) to cause that the agent executes the contribution and gets the speaker role (Figure 6.3.20 (G)).

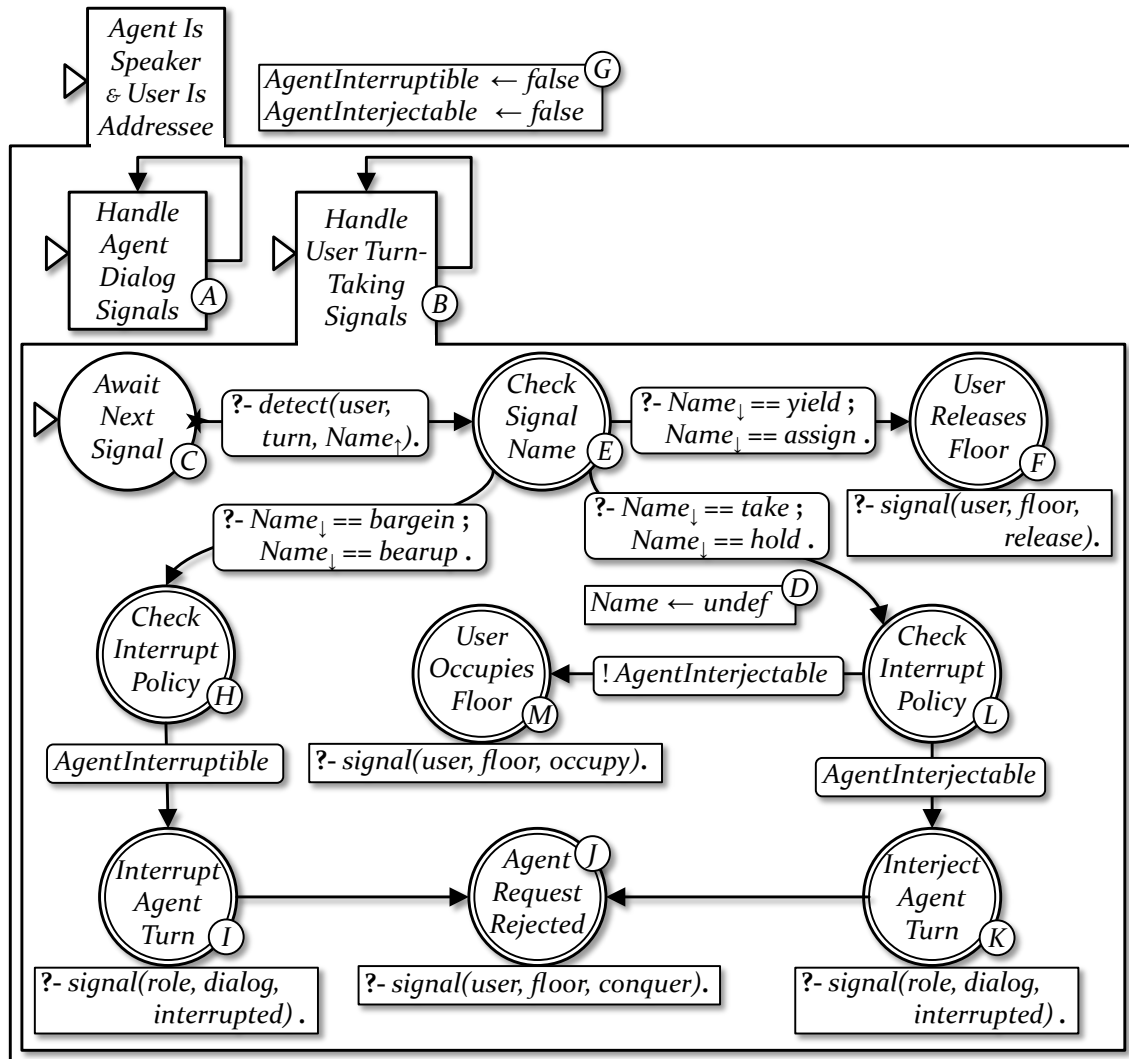


Figure 6.3.22: The behavior flow used for handling turn-taking and dialog signals as speaker.

THE AGENT
INTERRUPTION
POLICY

Figure 6.3.22 shows the behavior flow that decides if a role shift has to be performed while the agent is in the speaker role (Figure 6.3.20 (G)). A first nested behavior flow is handling the *dialog flow signals* (Figure 6.3.22 (A)) while the second behavior flow is processing the *turn regulation signals* (Figure 6.3.22 (B)). It waits until it detects a *turn action signal* or *turn conflict signal* in the fact base (Figure 6.3.22 (C)), extracts its name to the local variable *Name*

(Figure 6.3.22 ④) and afterwards checks which action or conflict may have caused the signal (Figure 6.3.22 ⑤). In case of a take or hold action, it checks if the agent is interruptible by this type of action (Figure 6.3.22 ⑥) and produces a *floor occupy signal* (Figure 6.3.22 ⑦) if the agent is not interruptible which causes that the user immediately acquires the speaker role when it is offered by the agent. In case of a yield or assign action, it produces a *floor release signal* (Figure 6.3.22 ⑧) which overrides a preceding *floor occupy signal*. In case of a barge-in or bear-up, the behavior flow also checks if the agent is interruptible by this conflict (Figure 6.3.22 ⑨). If the agent is interruptible, then it produces a *dialog interrupted signal* to cause that the agent interrupts itself (Figure 6.3.22 ⑩) and afterwards a *floor conquer signal* to cause an instant shift of the speaker role to the user (Figure 6.3.22 ⑪).

6.3.4 Dialog and Behavior Control

The *dialog and behavior control layer* (Figure 6.3.1 ⑥) comprises behavior flows that manage the dialog flow as well as the role-specific and role-independent aspects of the agent's non-verbal behavior. They finally control and produce the agent's observable deliberative and automatic behaviors and actions. They are closely synchronized with the participant role management layer (Figure 6.3.1 ⑦) and exchange information about the agent's and user's contributions with the dialog planning component. Furthermore, they are informed by signals produced by the input event handling layer (Figure 6.3.1 ④) and the behavioral pattern recognition layer (Figure 6.3.1 ⑤). This shows that most behaviors contributing to interpersonal coordination and grounding ultimately arise from the complex interplay and close coordination of the multiple processes in these layers of the model.

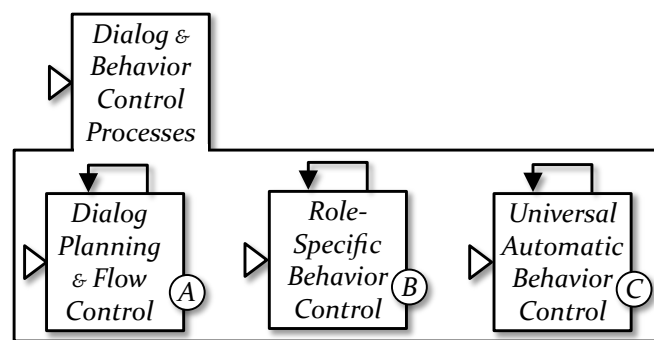


Figure 6.3.23: The behavior flow used for controlling the dialog flow and other behavioral aspects.

Figure 6.3.23 shows the hierarchical and parallel structure of the layer's main behavior flow. A first nested behavior flow is primarily controlling the dialog flow by coordinating the dialog planner with the participant role management (Figure 6.3.23 ①). Another one is producing the agent's role-specific nonverbal behavior based on the participant role assignments (Figure 6.3.23 ②). The third performs universal, that means role-independent, behavioral functions (Figure 6.3.23 ③), such as automatic idle head and body movements and postures, breathing and blinking behaviors, or physiological reactions. For reasons of redundancy, the remainder of this section only explains the first two nested behavior flows in more detail.

Dialog Planning and Flow Control

Figure 6.3.24 shows an extract of the behavior flow that models the dialog flow control. This task comprises the coordination with the planning component, the proper execution of the agent’s contributions as well as the handling of the user’s interruptions. It exchanges information with the dialog planner using *dialog planning signals* and is coordinated with the participant role management layer (Figure 6.3.1 (F)) via *dialog flow signals*. The dialog and behavior control model, as described here, doesn’t make any specific assumptions about the dialog planner. It only requires that the dialog planner is continuously and asynchronously planning the agent’s deliberative dialog contributions, that means instructions and clarification statements, in reaction to the user’s, contributions, which are questions and object movements. It is requesting the execution of the agent’s dialog contributions while the dialog and behavior model forwards the user’s speech acts and move actions in the opposite direction. The decisions made on the participant role management layer determine whether the agent’s planned dialog contributions may be executed, must be interrupted or have to be rejected and postponed already before. The dialog planner may be any suitable external software module, such as a rule-, plan-based, or statistical discourse planning engine (Rich and Sidner, 1998; Bohus and Rudnicky, 2003; Nooraei et al., 2014; Ultes and Minker, 2014) but may also be replaced by a *Wizard-of-Oz* interface operated by a human expert. Finally, it can also be realized with a specifically modeled parallel behavior flow whose structure implements the dialog’s branching logic, as already done in several other applications (Mehlmann et al., 2011b; Gebhard et al., 2012; Mehlmann et al., 2014a).

INTERRUPT & CONTRIBUTION HANDLING The first nested behavior flow in Figure 6.3.24 is managing the exchange and execution of contributions (Figure 6.3.24 (A)). It starts by waiting for the planning component to produce the agent’s next contribution while constantly processing the user’s contributions in a nested behavior flow (Figure 6.3.24 (B)). These are the user’s speech action signals produced when the user asks a question and the move action signals created when the user moves a puzzle piece on the surface table. We assume that they are simply propagated to the dialog planning component to update the dialog engine’s information state and discourse history and consider them for the next planning iterations.

When the dialog planner signals a new contribution, then the behavior flow extracts the relevant information (Figure 6.3.24 (C)) to the local variable *AgentContribution* (Figure 6.3.24 (D)). Afterwards, the participant role management layer is requested for the speaker floor by producing a *dialog request signal* (Figure 6.3.24 (E)) before waiting for the response signal (Figure 6.3.24 (F)). If the execution request is rejected, then the dialog planner is notified about the unsuccessful contribution attempt by propagating the respective *dialog rejected signal* (Figure 6.3.24 (G)) before restarting the behavior flow again (Figure 6.3.24 (B)). Otherwise, if the speaker role has been obtained, then the dialog planner is notified about the successful request by forwarding the corresponding *dialog accepted signal* (Figure 6.3.24 (H)) before the agent’s contribution is executed by the nested behavior flow (Figure 6.3.24 (I)), shown in Figure 6.3.25, and afterwards started over again (Figure 6.3.24 (B)).

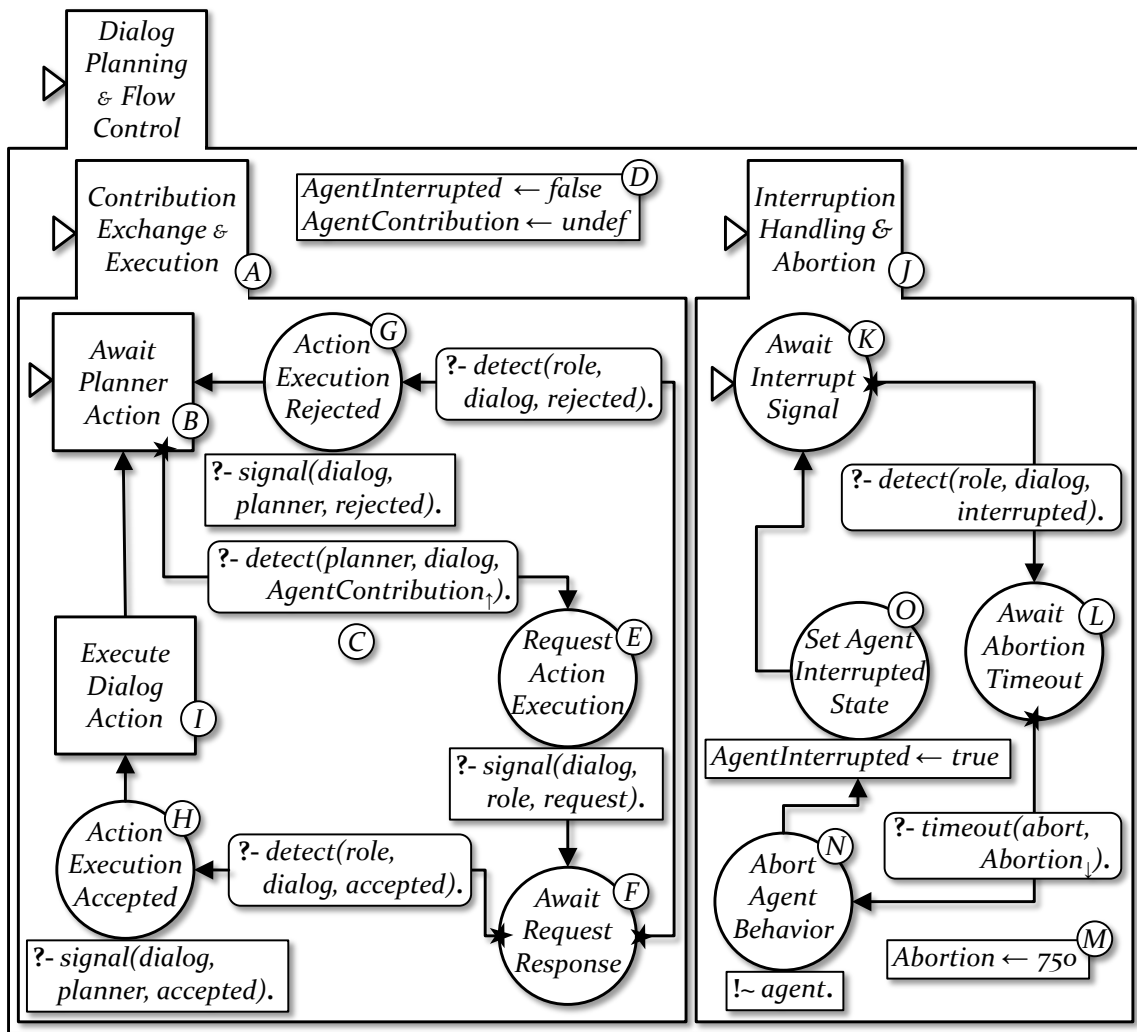


Figure 6.3.24: The behavior flow used for coordinating the dialog flow with the dialog planning.

In parallel to the dialog contribution logic, a second nested behavior flow is handling the user's interruptions and the abortion of the agent's behavior (Figure 6.3.24 J). It constantly waits for *dialog interrupted signals* from the participant role management layer (Figure 6.3.24 K). If such a signal is detected, then it awaits a short *abortion timeout* (Figure 6.3.24 L), defined by the local variable *Abortion* (Figure 6.3.24 M), before stopping the agent's utterance and all co-verbal behaviors (Figure 6.3.24 N). Afterwards, it instructs the parallel behavior flow (Figure 6.3.24 I) which is currently executing the agent's contribution to abort this execution with immediate effect (Figure 6.3.24 O). This synchronization is realized via the corresponding shared local variable *AgentInterrupted* (Figure 6.3.24 D).

Figure 6.3.25 shows the aforementioned behavior flow that controls the execution states and abortion of the agent's dialog contributions (Figure 6.3.24 J). Most interpersonal coordination and grounding behaviors in the speaker role, such as, for example, gaze aversion while planning speech, mutual gaze to the addressee when taking the turn, gaze behaviors revealing the cognitive or emotional state, mutual gaze to demand attention and directed gaze or

EXECUTION
SEQUENCE
& ABORTION

gestures to draw attention, and many more, are realized using appropriate activity specifications for the agent's verbal and co-verbal behavior in the scenes that are provided with the contribution and played back in consecutive states. In this specific case, a contribution consists of three scenes whose names are stored in the variables *DialogScene*, *AbortScene* and *AssignScene* (Figure 6.3.25 A). The *dialog scene* contains the agent's multi-modal utterance, the *abort scene* is executed if the dialog scene has been interrupted, and, finally, the *assign scene* is used to generate the appropriate nonverbal behaviors when assigning the turn to the user after the dialog scene has been played back.

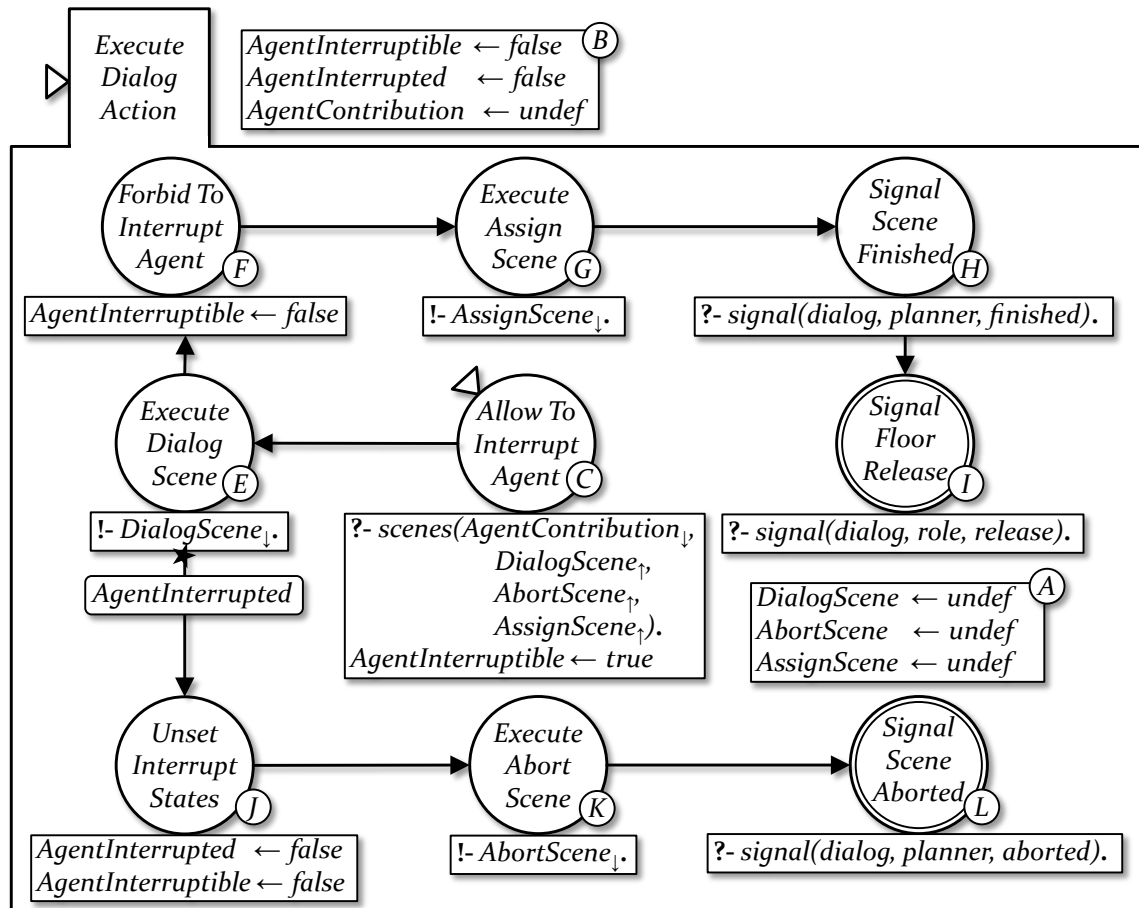


Figure 6.3.25: The behavior flow used for executing the agent's new contributions to the dialog.

The behavior flow first extracts the scenes from the contribution and then sets the global variable *AgentInterruptible* (Figure 6.3.25 B) to determine that the agent's next dialog scene must be *interruptible* (Figure 6.3.25 C). The decision to make the scene interruptible or not could, however, also be provided as part of the contribution in order to represent its urgency or importance. Then the *dialog scene* is played back (Figure 6.3.25 E) and the agent is set non-interruptible again after the scene has finished regularly (Figure 6.3.25 F). In this case, first, the *assign scene* is played back (Figure 6.3.25 G), then a *dialog finished signal* is produced to notify the dialog planner about the successful playback (Figure 6.3.25 H), and a *dialog release signal* is created to cause the participant role management to offer the

speaker role (Figure 6.3.25 ①). Otherwise, if the dialog scene is *interrupted*, then the variables *AgentInterrupted* and *AgentInterruptible* are reset (Figure 6.3.25 ②), the *abortion scene* is executed (Figure 6.3.25 ③), and, finally, the dialog planner is notified that the contribution was aborted by producing an *dialog aborted signal* (Figure 6.3.25 ④).

Role-Specific Behavior Control

Figure 6.3.26 shows the behavior flow that is used for the generation of the agent's role-specific, ideomotor, nonverbal behaviors. These behaviors fulfill essential functions for interpersonal coordination and grounding, such as, for example, attention following, back-channel production and intimacy regulation or the reaction to connection events and feedback eliciting cues. These functions can significantly differ between the individual participant roles, that means, the user's behaviors that are particularly responded in the one role can be handled completely differently or are even totally ignored in another role. Therefore, it is hierarchically structured such that each individual participant role is realized with a separate nested behavior flow producing exactly those behaviors that are typical for this specific role. Its execution is switching between the nested behavior flows based on the decisions made by the participant role management. In its current form, the addressee and speaker roles are not further refined, however, the model can be further refined such that also the transition phases that are entered whenever a participant is offering the speaker role (Figure 6.3.20 ④, ⑤) can as well be represented as separate nested behavior flows.

The behavior flow starts by executing the typical *overhearer* behavior while neither the user nor the agent are actively taking part in the interaction (Figure 6.3.26 ①). When both participants are involved in the interaction (Figure 6.3.26 ②), then the nested behavior flows are controlling the agent's automatic behavior in the *speaker* (Figure 6.3.26 ③), *bystander* (Figure 6.3.26 ④) or *addressee* role (Figure 6.3.26 ⑤). They produce the aforementioned role-dependent coordination and grounding behaviors that can easily be adapted to realize specific strategies or variations for a particular application. Transitions between the role-specific behavior flows are induced by the changes of the global variable *AgentRole* which is updated by the participant role management whenever a role decision has been made.

As representative example, Figure 6.3.26 shows the nested behavior flow which is used to model the agent's behavior in the *addressee* role in more detail (Figure 6.3.26 ⑤). It starts in a state in which the agent shows typical role-specific ideomotor listening behavior including, for example, irregular back-channels and probabilistic gaze distributions (Nielsen, 1962; Argyle and Ingham, 1972; Argyle *et al.*, 1973; Argyle and Cook, 1976; Argyle and Graham, 1976; Bee *et al.*, 2010b) (Figure 6.3.26 ⑥). For example, as suggested by Srinivasan *et al.* (2014), the agent could look at the user and sporadically look to randomly chosen positions in the environment for time periods between about 750 and 1000 milliseconds. Another possibility would be to implement the gaze distributions reported by Fukayama *et al.* (2002), Bee *et al.* (2010b), or Mutlu *et al.* (2012) to convey a particular impression or social attitude.

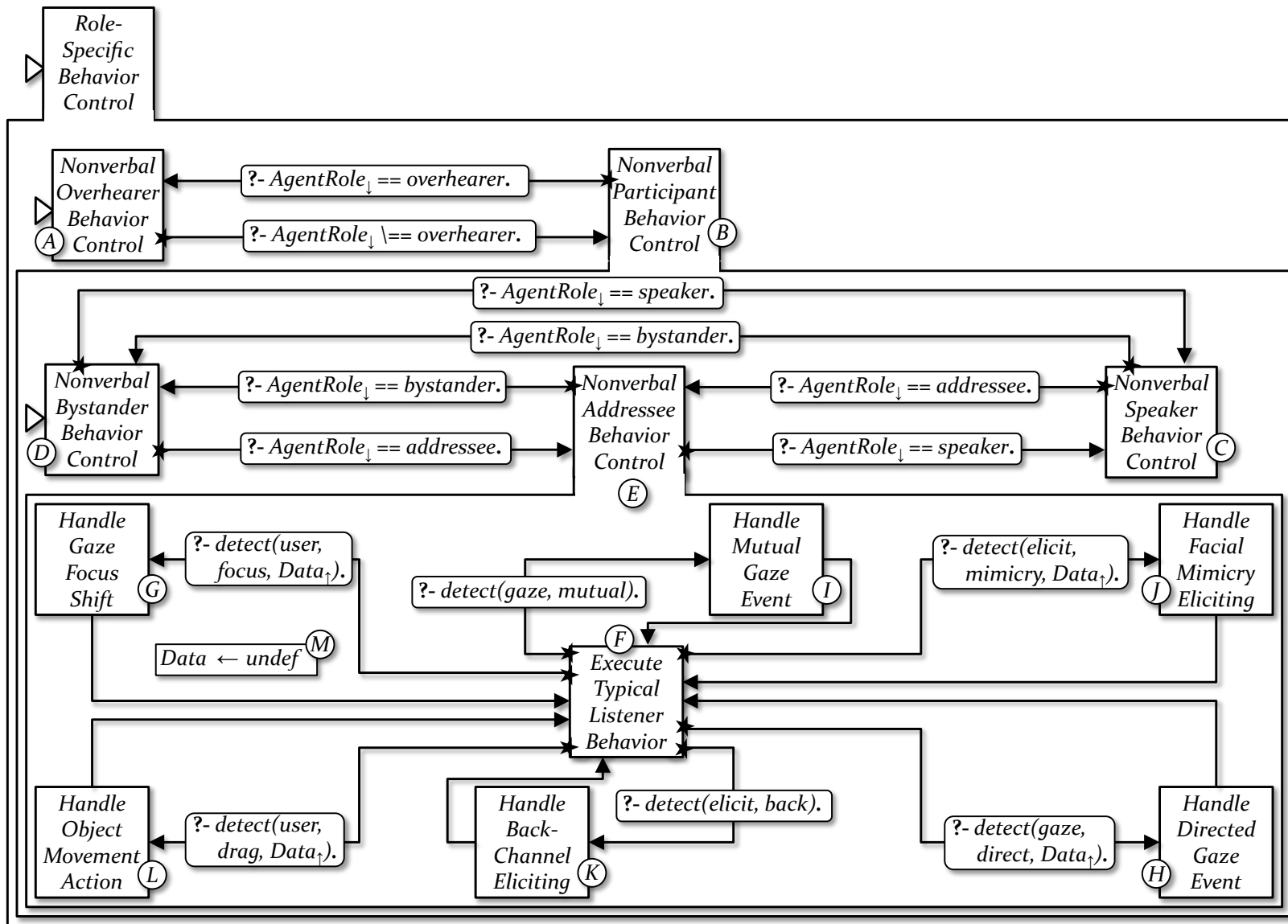


Figure 6.3.26: The behavior flow controlling the agent's role-specific nonverbal behavior based on the signals from the lower layers of the model.

Whenever the behavior flow is notified about a recognized behavioral pattern or user action by the lower layers of the model, then it reacts appropriately and afterwards returns to this listening behavior (Figure 6.3.26 ©-Ⓚ). The semantic information that is carried by the corresponding signals is stored in the local variable *Data* (Figure 6.3.26 Ⓜ) in order to be further processed by the nested behavior flows. For example, when the user changes his focus of visual attention, then the agent can adequately react in the corresponding nested behavior flow (Figure 6.3.26 ©). The agent could, for example, follow the user's gaze to the respective puzzle piece or answer the user's attempt to establish mutual gaze. The agent is also notified whenever he successfully established shared attention (Figure 6.3.26 Ⓜ) or has reached mutual facial gaze with the user (Figure 6.3.26 Ⓛ) and can then adequately react to these gaze connection events in the respective nested behavior flows. He could, for example, smile at the user when mutual facial gaze has been established or show interest by performing attentive facial expressions when they share the same perceptual ground during shared gaze. The agent is as well notified when the user is trying to elicit a facial mimicry (Figure 6.3.26 Ⓛ), such that he can, for example, respond to this eliciting cue with an emotional display in order to create the impression of emotional contagion or avert the gaze in order to balance the interpersonal intimacy. Furthermore, the agent is also notified when the agent tries to elicit a back-channel cue with a short glance of gaze while he is claiming the turn (Figure 6.3.26 Ⓚ). The decision whether or which type of back-channel should be produced could then be based on semantic information, for example, if the currently dragged object is the one that has been referred to in the agent's previous instruction. Finally, the agent is also constantly notified while the user moves a puzzle piece on the surface table, such that the agent is, for example, able to follow these object movements with his gaze in order to share the perceptual ground with the user (Figure 6.3.26 Ⓛ). It can clearly be seen, that this behavior control flow, that is informed by the behavior recognition and participant management layers of the model, offers a myriad of possibilities now to customize the coordination and grounding behaviors of the agent based on many influencing factors, such as models of politeness (Brown and Levinson, 1987), interpersonal relationships, or personality traits (Zimmerman and West, 1975; Ferguson, 1977; Natale *et al.*, 1979; Goldberg, 1990).

6.4 Summary and Conclusion

In this chapter, I illustrated the modeling approach proposed in this thesis using an exemplary behavior and interaction model of a social agent in an exemplary application. First, in Section 6.1, I presented the general application and sensor setup which is common to the demonstrator applications developed in this thesis and is also assumed for the exemplary application. Afterwards, in Section 6.2, I showed how the application-specific domain knowledge and the user's input events are represented in the application using feature structures in a *PROLOG* fact base. I especially explained how ambiguous referring expressions in the user's clarification requests can be disambiguated with gaze events during the respective speech event by using a logic quantification predicate. Afterwards, I present a large part of the *BFSC* modeling the interactive behavior of the social agent in the application. It is divided into

synchronized levels and processes for input handling and preprocessing, behavioral pattern recognition, participant role and turn management, and dialog and behavior control. The behavior and interaction model is assumed to work asynchronously together with an arbitrary dialog management module.

The presented model's architecture is highly modular using a multitude of parallel and nested behavioral levels and functions. It is rather generic and adaptable to application-specific needs. Thus it can serve as best practice example and toolbox resource of individual reusable and adaptable parts. The model includes most of the behavioral aspects that contribute to interpersonal coordination and grounding, such as attention following, multi-modal disambiguation, turn-taking, interruption handling. Thus, with regard to expressiveness, it certainly implements a superset of the capabilities of all other models for these behavioral functions that have been presented in Chapter 4. In fact, the model can easily be parameterized to resemble the behavior of those models, but, goes way beyond by combining their functionalities and their interplay in a single model.

PART IV

REALIZATION AND CONCLUSION

*“You should better walk your dog so that it
doesn't pee on the leash.”*

DR. ULRICH MEHLMANN

REALIZATION — RE-ENGINEERING AND VALIDATING THE VSM^3 FRAMEWORK

In Chapters 5 and 6 I presented and illustrated the theoretical foundations and conceptual framework of the *BFML* ensemble. Therefore, I explained how typed feature structures and a first- and higher-order logic calculus form the basis of an embedded domain-specific language in *PROLOG*. I introduced a specially designed state-chart dialect that supports the hierarchical refinement and parallel decomposition of the model as well as interruption policies and an exhaustive history mechanism. I finally defined a descriptive, template-based language to specify social agents' behavior and dialog similar to natural language scene scripts. I decided to depict the illustrative models in Chapters 5 and 6 in this conceptual notation in order to comprehensively illustrate these modeling concepts while waiving any “*syntactic sugar*”. However, I did not discuss their execution semantics such that the question how the conceptual syntax can be turned into an executable specification remained unanswered.

For that reason, this chapter now takes a closer look at the specification of the fully fledged, visual programming language that has been developed in this thesis for the most recent version of VSM^3 . It moves from the conceptual notation, given by the *BFML* ensemble, to an executable syntax defined in form of a revised and extended *Scene Flow Modeling Language* (*SFML*). I explain how the extension and adaptation of the former *scene flows*, used in the predecessor versions of VSM^3 (Rist *et al.*, 2002; Baldes *et al.*, 2002; Rist *et al.*, 2003; Klesen *et al.*, 2003; Gebhard *et al.*, 2003b; Gebhard and Klesen, 2005; Ndiaye *et al.*, 2005; Gebhard *et al.*, 2008; Schröder *et al.*, 2008; Mehlmann, 2009), led to the design of *extended scene flows* as reference implementation of the behavior flows presented in Chapters 5 and 6.

In the remainder of this chapter, in Section 7.1, I explain the redefinition of selected parts of the *SFML* specification, necessary to implement the conceptual framework from Chapter 5 in the latest version of VSM^3 . Afterwards, in Section 7.2, I present important refactoring steps that have been unavoidable to adapt the software architecture and components of VSM^3 to the specification's adjustment. Finally, in Section 7.3, I present a few demonstrator applications that have been developed in this thesis to validate the re-engineered version of VSM^3 .

7.1.1 The Specification of SFSCs

Re-engineering *VSM*³ included the re-design of *SFSCs* in order to integrate the modeling concepts of *BFSCs* described in Section 5.4. Like *BFSCs*, the major strength of the revised *SFSCs* is modeling the incremental and reciprocal interplay of processes for behavioral pattern recognition, knowledge reasoning, and behavior generation. They use parallel and hierarchical decomposition to make the model less complex, better readable, maintainable, and reusable. They rely on synchronization and information exchange means of the redesigned *SFGL* and *SFQL* to coordinate parallel behavioral processes, levels, functions, and modalities that contribute to interpersonal coordination and grounding behaviors of social agents.

The canonical *abstract syntax* of *SFSCs* can be mapped to different *textual syntaxes* that are suitable for *serialization* and modeling if no graphical editor is available. However, being first and foremost a *visual programming language*, the *visual syntax* of *SFSCs* is the most legible way to depict them and is, in the following, presented by means of the graphical notation of nodes, edges, and other visual characteristics of *SFSCs*. The meaning of these syntactical constructs is only insofar explained as it is necessary to get a rough idea how an operational execution semantics could structurally be defined (Harel and Naamad, 1996; von der Beeck, 1994; Drusinsky, 2004; Harel and Kugler, 2004; Drusinsky, 2006; Mehlmann, 2009).

Scene Flow Nodes

A node represents a small executable program segment consisting of *SFGL* statements. Thus, nodes are similar to functions in well-known procedural programming languages. Together with the guarding expressions of the edges connecting the nodes, the entire *SFSC* constitutes a static definition of a larger program whose semantics is the set of potential traces, which are the sequences of statements and transitions during the possible execution runs.

Basic Nodes The simplest type of node is a *basic node* which is historically also called *scene node* (Gebhard *et al.*, 2003a, 2008, 2012). It is graphically depicted with a circle which must be labeled with a name that is optionally followed by a unique identifier. Besides these two mandatory attributes, it can have additional visual features and be annotated with *SFGL* statements. Figure 7.1.2 shows the visual syntax of a basic node with the name *WelcomeScene* and the identifier N_1 . A reddish triangle (▶) marks it as regular *start node* of its parent node, thus serving as a starting point for the parent node's execution. Furthermore, the thin inner black circle (◦) inside the thicker grayish boundary of the node (○) marks it as *end node* which is a possible termination point of its executing process. Besides these visual attributes, the node is labeled with *SFGL* statements which are partitioned in different framed boxes above and below the node. The first box contains definitions of record-like data types, the second box contains variable definitions, and the third contains commands to be executed.

Super Nodes As shown in Figure 7.1.3, a *super node* is graphically represented by a square and can, just like a basic node, be labeled with the aforementioned visual features and *SFGL* statements. It additionally extends the functionality of a basic node because it can con-

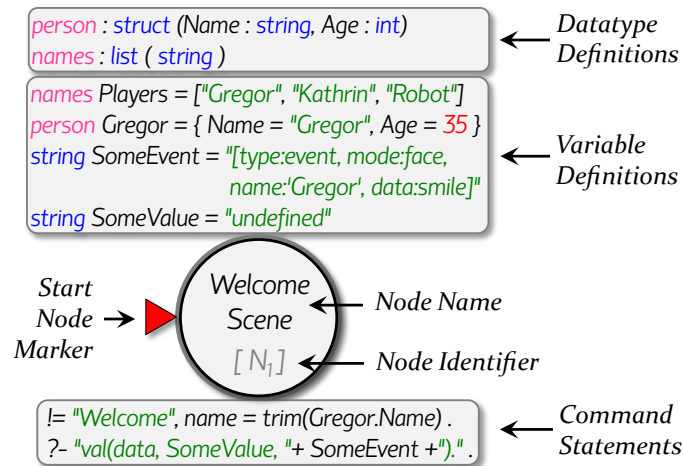


Figure 7.1.2: The visual syntax of a basic node with a few local definitions and command statements.

tain an arbitrary number of nested and concurrent *SFSCs*, thus creating a hierarchical and parallel structure. It may define any set of subnodes as regular start nodes, using reddish triangles (▶), to determine the parallel starting points of its execution. Each super node has a mandatory *history node* which is a special start node marked with a grayish triangle (▶). All definitions of types, variables, class paths, and functions of the super node are inherited by its nested subnodes. An incoming edge may specify *alternative start nodes* which then replace the other start and history nodes as execution starting points whenever the super node is entered via this edge. This is a way of parameterizing the execution of a super node depending on the preceding control flow. Different incoming edges may define differing alternative start node sets whose nodes are then marked with bluish triangles (▶).

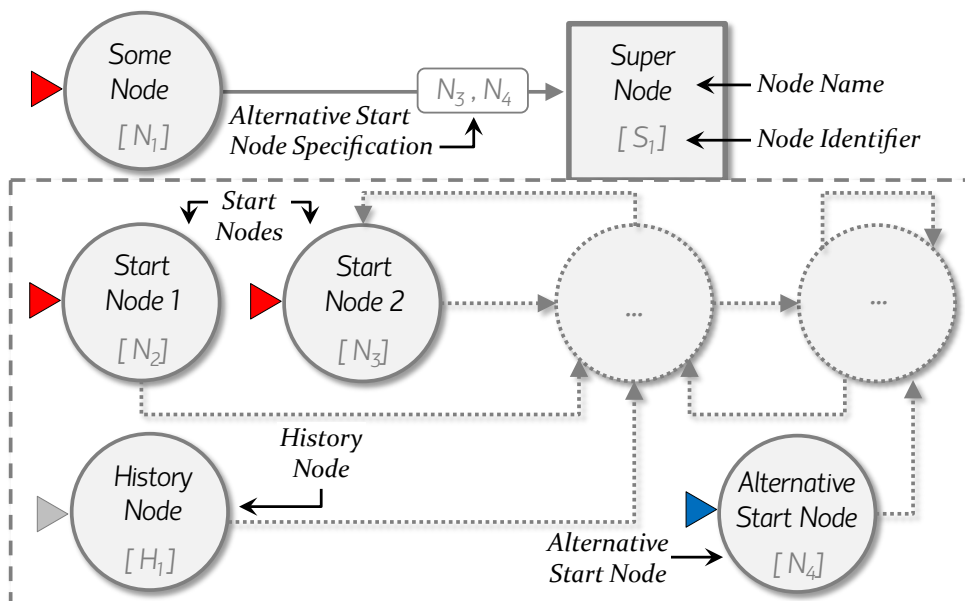


Figure 7.1.3: The visual syntax of a super node with a history node, regular and alternative start nodes.

Scene Flow Edges

SFSCs have different *edge types* that share a few common syntactical and semantical characteristics. They are directed, colored and optionally labeled arrows that are connecting two nodes, referred to as *source node* and *target node*. Edges can be labeled with different *guarding constraints*, that need to be satisfied to enable the transition. Like in BFSCs, these differ in when and by which process they are evaluated and how they come into effect.

Epsilon Edges An *epsilon edge* denotes an unconditional transition which is immediately taken when the execution of its source node is finished. As shown in Figure 7.1.4, epsilon edges are colored grayish and are unlabeled except they specify alternative start nodes. Their source nodes are also colored grayish if the epsilon edge is the only outgoing edge. They are used to create sequential structures and to determine the order of computation steps, such as the playback of behavioral activities or the execution of logic queries. They can make the model more clearly arranged and facilitate its manageability and readability.



Figure 7.1.4: An exemplary *epsilon edge* which is connecting the source node N_1 with target node N_2 .

Timeout Edges A *timeout edge* represents a timed transition that is taken with some delay when the timeout has expired after the execution of the source node. As shown in Figure 7.1.5, timeout edges are colored brownish and are labeled with a timeout value in milliseconds. Their source nodes are as well colored brownish if they have only outgoing timeout edges. They are used to control the timing and scheduling of consecutive computation steps.

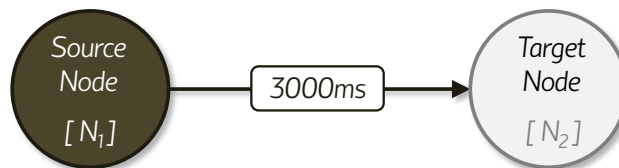


Figure 7.1.5: An exemplary *timeout edge* which is connecting the source node N_1 with target node N_2 .

Probability Edges A *probability edge* denotes a transition that is taken with the specified probability after the execution of the source node has finished. As shown in Figure 7.1.6, probability edges are colored greenish and labeled with a percentage value between zero and one hundred. Their source nodes are also colored greenish and may have exclusively outgoing probability edges whose probabilities must sum up to 100% to cover the entire probability space. They are used to create a certain degree of randomness in the model's branching structure and thus a desired non-deterministic behavior.

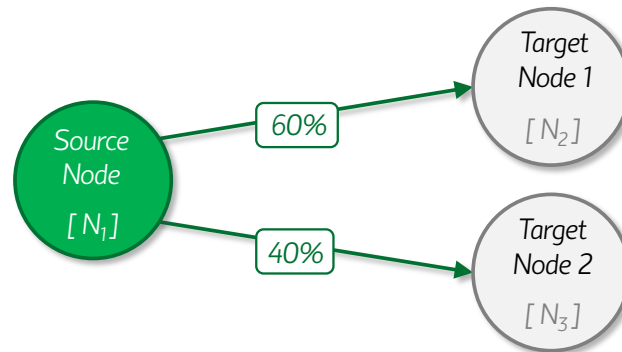


Figure 7.1.6: Two exemplary *probability edges* from the source node N_1 to the target nodes N_2 and N_3 .

Condition Edges A *condition edge* represents a conditional transition that is enabled when the guarding constraint is satisfied after the execution of the source node has finished. As shown in Figure 7.1.7, condition edges are colored orange and labeled with a guarding conditional or query expression. Their source nodes are also colored orange if they have solely additional outgoing epsilon or timeout edges. They are used to create a conditional branching structure and to determine the reaction to user inputs and context changes.

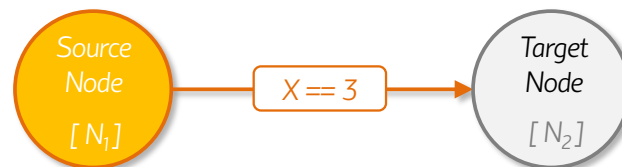


Figure 7.1.7: An exemplary *condition edge* that is connecting the source node N_1 with target node N_2 .

Interruptive Edges An *interruptive edge* represents a conditional transition that, in contrast to a regular condition edge, taken “*immediately*” when its guarding constraint is satisfied even if the execution of the source node has not yet finished. In this case, the source node’s execution is interrupted and the nested nodes’ execution is terminated in the same step. Therefore, the currently executed activities such as the playback of behavioral activities or the execution of functions have to be aborted and returned from. Consequently, interruptive edges at nodes closer to the root have priority over those farther from the root.

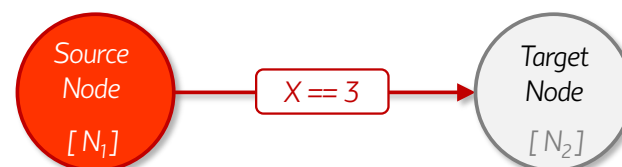


Figure 7.1.8: An exemplary *interruptive edge* connecting the source node N_1 with the target node N_2 .

As shown in Figure 7.1.8, interruptive edges are labeled with a guarding conditional or query expression and are, just like their source nodes, colored reddish. They are used to realize interruptions and priorities in response to user inputs or external events that require an immediate handling and behavioral reaction without undue delay.

Forking Edges A *forking edge* denotes an unconditional transition that starts a new parallel process when it is taken. As shown in Figure 7.1.9, forking edges are unlabeled and colored bluish like their source nodes that must not have outgoing edges of another type. With the help of forking edges the execution may be split into multiple concurrent processes on the same hierarchy level without the need to use super nodes with multiple start nodes.

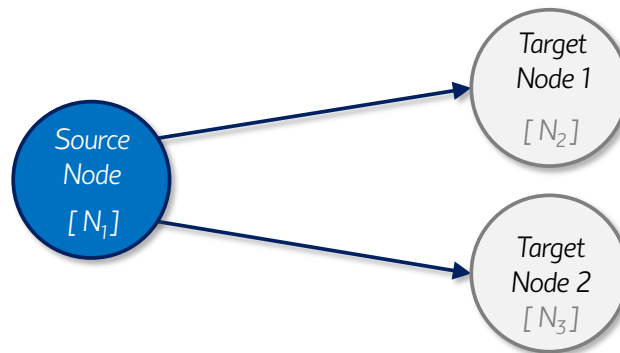


Figure 7.1.9: Exemplary forking edges connecting the source node N_1 with target nodes N_2 and N_3 .

7.1.2 The Specification of SFGL

A *glue language* is a programming language that is designed to write and manage code, which supports interconnecting different software components, programming languages, and platforms. As such, the responsibility of SFGL is to glue together SFSCs with the underlying implementation language JAVA™ and the other ensemble members of SFML. It is used to label the nodes of SFSCs with definitions, assignments, invocations, and expressions, such as the playback of SFSL specifications, the evaluation of SFQL queries, and the execution of JAVA™ functions. Furthermore, it is used to label the edges of SFSCs with transition guards, such as arithmetical and logical conditions, or SFQL query expressions. JAVA™ with its *reflection mechanism* (Forman and Forman, 2004) is used to integrate native libraries and software components for sophisticated tasks whose realization would be too laborious with SFML.

Similar to the textual expression languages of other state-chart dialects (Harel, 1987; Harel et al., 1990; Harel and Naamad, 1996; Harel and Politi, 1998; von der Beeck, 1994; Drusinsky, 2004; Harel and Kugler, 2004; Drusinsky, 2006; Crane and Dingel, 2007), SFGL is a simple procedural and imperative scripting language, lexically and syntactically very close to other well-known glue languages, such as JavaScript¹, Python², or Perl³. It resembles a subset of its implementation language JAVA™ with a notably simplified type system and considerably reduced expressiveness. The data interchange between the SFML ensemble members is realized via primitively or record-like typed variables while values are implicitly converted.

Re-engineering VSM³ required the revision of the SFGL to apply the BFGL modeling concepts described in Sections 5.2 to 5.4. Besides some less extensive lexical and syntactical simplifications, this included the extension with new commands for the playback of SFSL scenes and

¹<https://www.javascript.com/>

²<https://www.python.org/>

³<https://www.perl.org/>

actions as well as the evaluation of *SFQL* queries to the newly integrated *SWI-PROLOG* fact base. In the following, I illustrate these adjustments via selected parts of *SFGL*'s syntax using context-free grammar production rules in *Extended Backus-Naur Form (EBNF)* and *regular expressions* defining the lexemes of *SFGL*.

Command Types As shown in Listing 7.1.1, a command statement of *SFGL* can be a definition statement, a variable assignment, a method invocation, or an expression statement.

```
command →  
    definition  
    | assignment  
    | invocation  
    | expression
```

Listing 7.1.1: The redefined production rules for command types in the *EBNF* grammar of the *SFGL*.

Definition Types As shown in Listing 7.1.2, a definition statement of the *SFGL* can represent the definition of a new data type, variable, function, or class path.

```
definition →  
    datatype_definition  
    | variable_definition  
    | function_definition  
    | classpath_definition
```

Listing 7.1.2: The redefined production rules for definition types in the *EBNF* grammar of the *SFGL*.

Type Definitions As shown in Listing 7.1.3, a data type definition introduces a new type name for a record-like list or struct type.

```
datatype_definition →  
    list_type_definition  
    | struct_type_definition  
  
list_type_definition →  
    datatype_name : list ( primitive_type )  
  
struct_type_definition →  
    datatype_name : struct ( opt_member_definition_list )  
  
opt_member_definition_list →  
  
    | member_definition_list  
  
member_definition_list →  
    member_definition
```

```

| member_definition_list , member_definition

member_definition →
    identifier : primitive_type

primitive_type →
    int | short | long | float | double | bool | char | string

datatype_name →
    ([ a-zA-Z ] | _ ) ([ a-zA-Z ] | [ 0-9 ] | _ ) *

```

Listing 7.1.3: The redefined production rules for type definitions in the *EBNF* grammar of the *SFGL*.

Variable Definitions As shown in Listing 7.1.4, a variable definition initializes a new typed variable with an expression.

```

variable_definition →
    type_identifier identifier = expression

type_identifier →
    primitive_type | datatype_name

identifier →
    ([ a-zA-Z ] | _ ) ([ a-zA-Z ] | [ 0-9 ] | _ ) *

```

Listing 7.1.4: The new production rules for variable definitions in the *EBNF* grammar of the *SFGL*.

Function Definitions As shown in Listing 7.1.4, a function definition introduces a new function name that refers to a *JAVA*[™] function which is specified by the name of a member function with its parameter list and the class path of the enclosing *JAVA*[™] class.

```

function_definition →
    identifier : fun ( class_qualifier , identifier )
| identifier : fun ( class_qualifier , identifier , parameter_definition_list )

parameter_definition_list →
    parameter_definition
| parameter_definition_list , parameter_definition

parameter_definition →
    identifier : class_qualifier

class_qualifier →
    identifier
| class_qualifier . identifier

```

Listing 7.1.5: The new production rules for function definitions in the *EBNF* grammar of the *SFGL*.

Class Path Definitions As shown in Listing 7.1.6, a class path definition consists of a class name alias and the qualified path to the referenced JAVA™ class.

```
classpath_definition →  
    identifier : class ( class_qualifier )
```

Listing 7.1.6: The new production rules for class path definitions in the EBNF grammar of the SFGL.

Variable Assignments As shown in Listing 7.1.7, a variable assignments assigns an expression to a variable expression.

```
assignment →  
    variable_expression = expression  
  
opt_assignment_list →  
  
    | assignment_list  
  
assignment_list →  
    assignment  
    | assignment_list , assignment
```

Listing 7.1.7: The new production rules for variable assignments in the EBNF grammar of the SFGL.

Method Invocations As shown in Listing 7.1.8, a method invocations can, among others, be an action and scene playback commands or a built-in history function.

```
invocation →  
    !~ expression .  
    | !~ expression , assignment_list .  
    | !- expression .  
    | !- expression , assignment_list .  
    | != expression .  
    | != expression , assignment_list .  
    | PlayScene ( expression )  
    | PlayScene ( expression , assignment_list )  
    | PlayAction ( expression )  
    | PlayAction ( expression , assignment_list )  
    | SetDefaultStrategy ( expression )  
    | SetDefaultLanguage ( expression )  
    | UnblockSceneScript ( boolean )  
    | UnblockSceneGroup ( expression )  
    | HistorySetDepth ( identifier , integer )  
    | HistoryFlatClear ( identifier )  
    | HistoryDeepClear ( identifier )
```

Listing 7.1.8: The new production rules for method invocations in the EBNF grammar of the SFGL.

Expressions Types As shown in Listing 7.1.9, an expression may be a unary, binary, ternary, invocation, reflection, variable, record, or literal expressions. The operator precedences and associativities in *SFGL* are the same as in *JAVA*[™] and other programming languages.

```

expression →
    unary_expression
  | binary_expression
  | ternary_expression
  | reflection_expression
  | invocation_expression
  | variable_expression
  | record_expression
  | literal_expression
  | ( expression )

unary_expression →
    unary_operator expression

binary_expression →
    expression binary_operator expression

unary_operator →
    ! | - | ~ | ++ | --

binary_operator →
    & | | | ^ | && | || | < | > | <= | >= | = | != | + | - | * | / | %

ternary_expression →
    ( expression ? expression : expression )

reflection_expression →
    identifier ( opt_expression_list )

opt_expression_list →
    | expression_list

expression_list →
    expression
  | expression_list , expression

```

Listing 7.1.9: The redefined production rules for expressions in the *EBNF* grammar of the *SFGL*.

Variable Expressions As shown in Listing 7.1.10, a variable expressions can denote a local or global variable, a value at a specific index of a list, or the member of a struct.

```

variable_expression →
    identifier

```

```

| identifier [ expression ]
| identifier . identifier

```

Listing 7.1.10: The new production rules for variable expressions in the *EBNF* grammar of the *SFGL*.

Invocation Expressions As shown in Listing 7.1.11, an invocation expressions can be a logic query as well as built-in history, timeout, random, and configuration expressions.

```

invocation_expression →
    ?- expression .
    | ?= expression .
    | Timeout ( expression )
    | Random ( expression )
    | InState ( identifier )
    | Contains ( expression, expression )
    | HistoryContains ( identifier , identifier )
    | HistoryContains ( identifier , identifier , integer )
    | HistoryValueOf ( identifier , identifier )
    | HistoryValueOf ( identifier , identifier , integer )
    | HistoryRunTimeOf ( identifier )
    | HistoryRunTimeOf ( identifier , integer )

```

Listing 7.1.11: The new production rules for invocation expressions in the *EBNF* grammar of the *SFGL*.

Records and Literals As shown in Listing 7.1.12 a record can be a list or struct and a literal can be a boolean, integer, float, or string.

```

record_expression →
    [ opt_expression_list ]
    | { opt_assignment_list }

literal_expression →
    boolean
    integer
    float
    string

boolean → true | false

integer → 0|-?[1-9][0-9]*

float → (0|-?[1-9][0-9]*).[0-9]+

string → "([a-zA-Z][0-9][\ \t][!?$&#/=~_.,\+*-%|()<>{}[]"@])*"

```

Listing 7.1.12: The new production rules for records and literals in the *EBNF* grammar of the *SFGL*.

7.1.3 The Specification of SFSL

Re-engineering *VSM*³ included the redefinition of *SFSL* in order to integrate the modeling concepts of the *BFSL* described in Section 5.2. This basically comprised the extension with the new types of behavioral activities and the new syntax of nested and standalone actions as well as scene activities and scene scripts. Like *BFSL*, the revised *SFSLs* can now rely on the full power of both scene and action activities for the creation of multi-modal behavior and dialog content which may be enriched with context information using parameters.

For redundancy reasons, I solely present the lexical and syntactical modifications of scene script definitions and abstain from showing the analogous adaptations for action activities. The syntax definitions for action and utterance activities are nearly identical to certain parts of the following scene syntax definitions. Therefore, I present only some selected parts of the *SFSL*'s syntax definitions, using production rules of a context-free grammar in *Extended Backus-Naur Form (EBNF)* and *regular expressions* that define the lexemes of *SFSL*. Figure 7.1.10 shows an exemplary scene definition in the redesigned syntax and labeling with the names of the most important language constructs explained in the following.

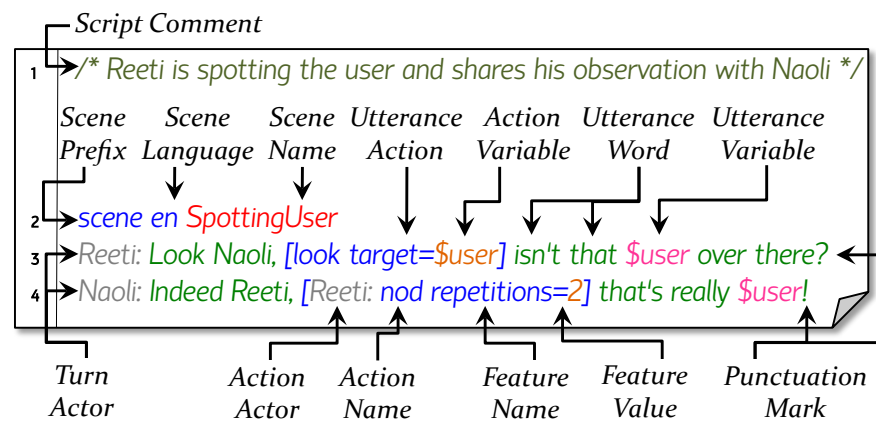


Figure 7.1.10: An exemplary scene script containing a comment and a single scene definition in *SFSL*.

Script Definitions As shown in Listing 7.1.13, a scene script is defined as an empty document or a non-empty list of script entity definitions, each of which can be a comment or a scene definition. A comment is a sequence of alphanumeric, whitespace, or newline characters that are delimited by an opening (*/**) and closing comment marker (**/*).

```

scene_script_definition  →
    opt_script_entity_list

opt_script_entity_list  →
    | script_entity_list

script_entity_list     →
    script_entity

```

```
| script_entity_list  script_entity

script_entity  →
    scene_definition
    | script_comment

script_comment → /* .* */
```

Listing 7.1.13: The new production rules for scene scripts in the *EBNF* grammar of the *SFSL*.

Scene Definitions As shown in Listing 7.1.14, a scene definition consists of a header and a content. The newline-ended header includes a scene keyword, a language and an identifier while the content consists of a list of turn definitions.

```
scene_definition  →
    scene_header scene_content

scene_header →
    scene_keyword scene_language scene_identifier newline

scene_keyword → scene | Scene

scene_language → [a-z]2 | [A-Z]2

scene_identifier → [a-zA-Z]([a-zA-Z]|[0-9]|_)*

scene_content →
    turn_definition_list newline

turn_definition_list →
    turn_definition
    | turn_definition_list turn_definition

newline → \r|\n|\r\n
```

Listing 7.1.14: The new production rules for scene definitions in the *EBNF* grammar of the *SFSL*.

Turn Definitions As shown in Listing 7.1.15, a turn definition is a newline-ended sequence consisting of a turn actor name, a colon mark, and a non-empty sequence of individual turn element definitions, which can, for example be utterance or pause definitions.

```
turn_definition  →
    turn_actor_name : turn_element_definition_list newline

turn_actor_name → [a-zA-Z]+

turn_element_definition_list →
```

```

    turn_element_definition
  | turn_element_definition_list turn_element_definition

turn_element_definition →
    utterance_definition
  | pause_definition

```

Listing 7.1.15: The new production rules for turn definitions in the *EBNF* grammar of the *SFSL*.

Utterance Definitions As shown in Listing 7.1.16, an utterance definition is a sequence of utterance elements that are separated by whitespaces but no newline characters and ended by a punctuation mark.

```

utterance_definition →
    utterance_element_list punctuation_mark

utterance_element_list →
    utterance_element
  | utterance_element_list utterance_element

punctuation_mark → . | ? | ! | , | ; | -

```

Listing 7.1.16: The new production rules for utterance definitions in the *EBNF* grammar of the *SFSL*.

Utterance Elements As shown in Listing 7.1.17, an utterance element can be an simple word, a nested action, or a placeholder variable.

```

utterance_element →
    utterance_word
  | utterance_action
  | utterance_variable

utterance_word →
    [a-zA-Z]+ | [1-9][0-9]* | [a-zA-Z]+'[a-zA-Z]

utterance_variable →
    $[a-zA-Z]([a-zA-Z]|[0-9]|_)*

```

Listing 7.1.17: The new production rules for utterance elements in the *EBNF* grammar of the *SFSL*.

Utterance Actions As shown in Listing 7.1.18, an utterance action is defined by an optional actor and a mandatory name, followed by an optional feature list.

```

utterance_action →
    [ actor_name opt_action_feature_list ]

```

```

| [ action_actor : action_name opt_action_feature_list ]

action_actor → [a-zA-Z]+

action_name → [a-zA-Z]([a-zA-Z] | [0-9] | _)*

opt_action_feature_list →

| action_feature_list

action_feature_list →
  action_feature
| action_feature_list action_feature

```

Listing 7.1.18: The new production rules for utterance actions in the *EBNF* grammar of the *SFSL*.

Action Features As shown in Listing 7.1.19, an action feature is a key value pair that consists of a feature name and a feature value which can be a primitive value, such as an identifier, boolean, integer, float, or single-quoted string as well as a parameter variable.

```

action_feature →
  feature_name = feature_value

feature_name → [a-zA-Z]([a-zA-Z] | [0-9] | _)*

feature_value →
  parameter
| identifier
| boolean
| integer
| float
| string

parameter → $[a-zA-Z]([a-zA-Z][0-9]|_)*

identifier → [a-zA-Z]([a-zA-Z][0-9]|_)*

boolean → true | false

integer → 0|-?[1-9][0-9]*

float → (0|-?[1-9][0-9]*).[0-9]+

string → '([a-zA-Z][0-9][\ \t][!?$&#/=~.,\+*-%|()<>{}]"@])*'

```

Listing 7.1.19: The new production rules for action features in the *EBNF* grammar of the *SFSL*.

7.1.4 The Integration of SFQL

Of course, re-engineering *VSM*³ also included the integration of the *SFQL* in order to realize the modeling concepts of *BFQL* described in Section 5.3. Like *BFQL* it is used for maintaining an event history with garbage collection as well as context and domain knowledge that is too complex to be represented as variables in the simple type system of *SFGL*. *SWI-PROLOG* queries are handed over using the aforementioned query expressions of *SFGL* and are used for multi-modal fusion and reasoning on the fact base or retrieving information and extracting it to *SFGL* variables for further processing in the *SFSC*. As mentioned before, *SFQL* is the *SWI-PROLOG* implementation of the standard *PROLOG*-based *BFQL* and thus differs only marginally from *BFQL*. Consequently, the semantics of the queries called via *SFGL* are precisely those of *SWI-PROLOG*. It consists of a hierarchy of *SWI-PROLOG* modules each of which defines specific rules and facts using the standard *SWI-PROLOG* syntax. Figure 7.1.11 shows a diagram illustrating the *SWI-PROLOG* module structure and relationships of the *SFQL*. The complete *SWI-PROLOG* source code of *SFQL* can be found in the freely downloadable open-source version of *VSM*³ ⁴. The core components of *SFGL* may easily and straightforwardly be extended in a well-defined and semantically unambiguous way by including new application-specific *SWI-PROLOG* modules.

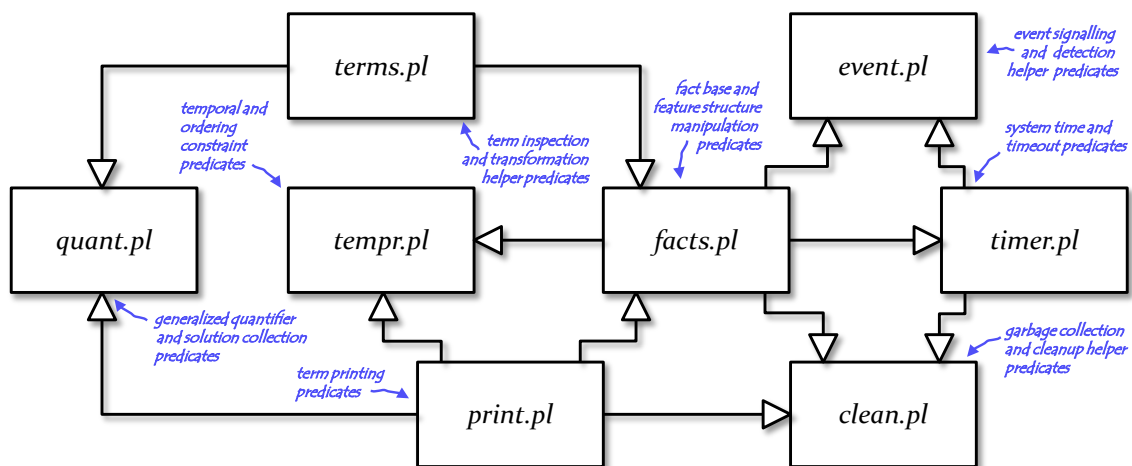


Figure 7.1.11: A diagram showing the *SWI-PROLOG* module structure and relationships of the *SFQL*.

7.2 Refactoring Software Components

The re-design and redefinition of *SFML* necessitated the substantial refactoring of *VSM*³'s software architecture and components. In this, the re-engineered *VSM*³ framework was divided into separated software and data layers which are depicted in Figure 7.2.1. The *modeling layer* (Figure 7.2.1 (A)) consists of an *Integrated Development Environment (IDE)* that may be complemented by an extern *SWI-PROLOG* editor ⁵. The *IDE* comes with a *graphical user interface* comprising several editor and configuration components. Among those, are an editor

⁴<http://scenemaker.dfki.de>

⁵<http://www.swi-prolog.org/IDE.html>

to visually model *SFSCs* and label them with *SFGL* statements and a textual *SFSL* editor with syntax checking and highlighting. The *IDE*, manages the models within *VSM³* project directories containing files with the respective *XML* representations and configurations. These data sources together make up the *data sources layer* (Figure 7.2.1 (B)) and may be parsed and written into *Abstract Syntax Trees (ASTs)* that are made up of classes of the *Data Model Definition (DMD)* on the *data model layer* (Figure 7.2.1 (C)). The *ASTs* of *SFSCs* are interpreted on the *runtime layer* (Figure 7.2.1 (D)) by components of the *Interpreter Runtime Environment (IRE)*, including an evaluator for *SFGL* expressions, a player for *SFSL* specifications, and a server for *SFQL* expressions. The runtime layer also defines the plug-in and executor interfaces that must be implemented by external components for integrating new agent platforms, output devices, and input sources on the *plug-in layer* (Figure 7.2.1 (E)).

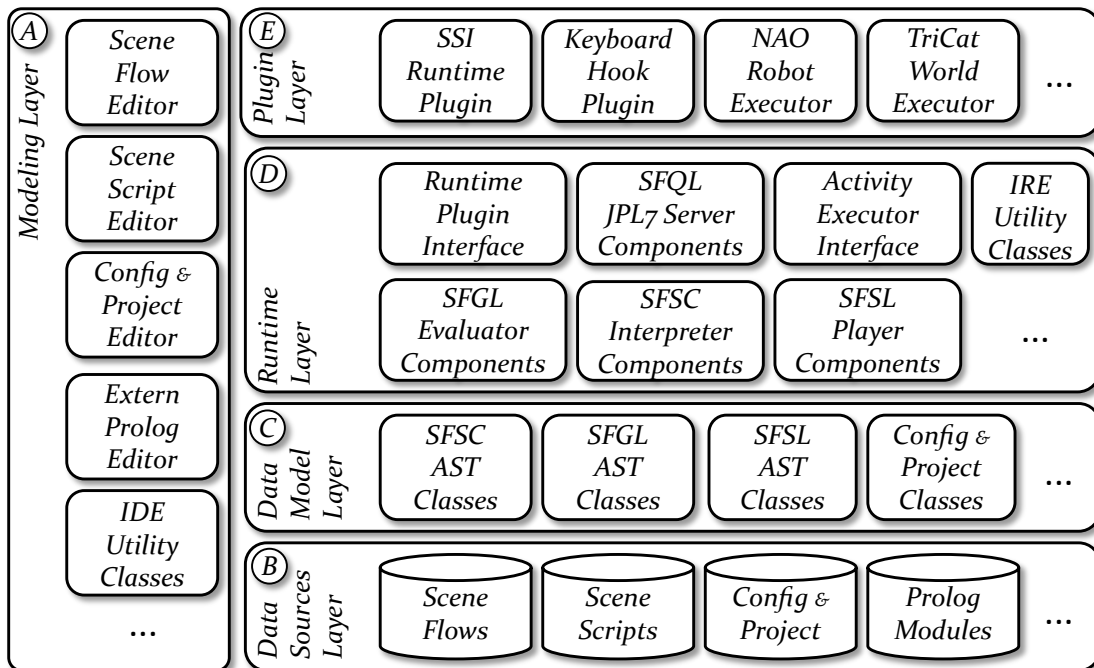


Figure 7.2.1: A diagram showing an overview of important software layers and components of *VSM³*.

In the following, I address a few important refactoring steps that have been applied to the aforementioned software layers and components. I firstly discuss some design considerations before I present selected parts of the *Data Model Definition (DMD)*, the *Interpreter Runtime Environment (IRE)*, and the *Integrated Development Environment (IDE)* of *VSM³*.

7.2.1 Design Considerations

Different aspects have to be considered when deciding for an implementation approach to a real-time capable interpreter component for *SFML* models. In this context, the term *real-time* means that the interpreter software has to be able to react to environmental events, user inputs as well as model modifications by an author in sufficient time. Scheduled actions have to be executed without much latency, according to the definition of a *soft real-time system*

(Rechenberg and Pomberger, 1997). In terms of *timeliness* and *reliability*, the interpreter software has to allow executing an extended scene flow so that the user has the impression of a natural human-like form of communication with timing conditions that are similar to those that occur in social human interactions. This section contains a comparison of an *interpretation* approach with a *compilation* approach to the implementation of an execution software for extended scene flows and motivates the decision to pursue the interpretation approach. Furthermore, there are discussed different techniques for the representation and scheduling of *multiple processes* in the implementation. Advantages and disadvantages of these techniques are compared in order to motivate the *multi-threaded* implementation of an *interpreter* software. Finally, there is justified the decision to choose *JAVA™* as the implementation language for the execution software in this work.

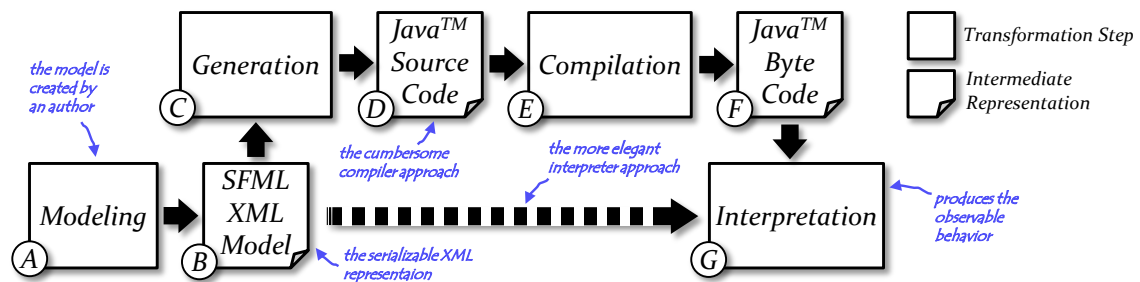


Figure 7.2.2: The sequence of steps performed by an author in order to finally execute a SFML model.

Interpreter vs. Compiler

As shown in the upper part of Figure 7.2.2, with the earliest versions of the authoring suite, an author had to carry out a multitude of consecutive steps in order to finally execute a model. After the modeling phase (Figure 7.2.2 (A)), the model's XML representation (Figure 7.2.2 (B)) had to be translated into *JAVA™* source code (Figure 7.2.2 (D)) using a source-to-source compiler (Figure 7.2.2 (C)). Then, the generated *JAVA™* classes had to be compiled to *JAVA™* byte-code (Figure 7.2.2 (F)) using a standard *JAVA™* compiler (Figure 7.2.2 (E)). Finally, the model was executed by interpreting (Figure 7.2.2 (G)) this machine-code representation with the *JAVA™* Virtual Machine. Using this *compilation approach* has the disadvantage that an author's modification of the model requires to repeat all these steps (Figure 7.2.2 (A)-(G)) in order to make the resultant effects observable during the execution. Since this proceeding is obviously not very comfortable and rather time consuming, *VSM³*, in its current form, pursues the direct *interpretation approach* for the execution of a SFML model, as shown on the bottom of Figure 7.2.2 (Figure 7.2.2 (A), (B), (G)). This interpreter approach has the advantage that modifications of the model are applied at runtime and the resultant effects can immediately be observed during the execution without the need to pause the execution, generate source code from the modified model and compile that source code again. It allows modifying scene definitions in the scene script, adjust nodes and edges in the scene flow or even exchange plug-in modules at runtime. This method gives an author a direct feedback of the undertaken modifications, supporting a rapid and comfortable prototyping process.

Single- vs. Multi-Threaded

EXECUTION
VIA PRODUCT
AUTOMATON

Since each parallel scene flow can be considered as an individual process, it is important how to represent and execute such a process in the aforementioned interpreter. In this light, each parallel scene flow may be regarded as *program graph*, treating each statement, or more generally, each step of the *interleaving semantics*, described in Section 5.4.1, as a transition statement in this program graph. The interleaving semantics then implicitly creates a *product automaton* of these program graphs which describes the overall behavior of the entire model. Using a *transformation function*, which compiles a parallel extended scene flow model into a semantically equivalent sequential counterpart, using program graphs and product automaton as intermediate representations, would allow using the single-threaded execution software of the early versions. However, this would require to fall back on the compiler approach for the execution of the model. In addition, on further reflection, it turns out that it is not a straightforward task to find an appropriate equivalence transformation. Consequently, this option was refused for the re-engineering of VSM³ and parallel scene flows are instead regarded as individual concurrent processes.

SINGLE-
THREADED
EXECUTION

Parallel processes can, on the one hand, be implemented by *multiple threads* that are provided by the underlying implementation language and operating system. On the other hand, the execution and scheduling of multiple parallel processes may be simulated within a *single thread* only (Jacobs and Verbraeck, 2004). The simulation approach enhances the control over the scheduling mechanism and avoids synchronization measures for mutual exclusion to critical sections which are necessary when using operating system threads. It enables an easy interruption and termination of processes by removing them from the set of scheduled processes and facilitates debugging since processes can be stopped or paused after each step to carry out a detailed analysis of the execution state. Being independent of the underlying operating system's thread and scheduler implementation, execution traces are reproducible and execution states are serializable such that they can be recoded, streamed over a network and stored to files. Finally, the number of processes is not restricted by the operating system and their execution might consume less time and space resources than system threads.

MULTI-
THREADED
EXECUTION

Scheduling the interleaving of simulated processes by hand can, however, also be considered as disadvantage. The *scheduling policy* must be *fair* in the sense that each processes has to be dealt to the same amount of computation resources or time and that there may never occur cases of starvation. Since all simulated processes are executed within a single thread of the operating system, it is basically impossible to achieve a fair scheduling between the simulated processes and external threads whose scheduling can not be brought under the control of the execution environment. Such threads could, for example, be used for the playback of behavioral activities, the execution of logic queries or the outsourcing of specific computations to function calls in the underlying implementation language. However, when falling back on the scheduler of the underlying operating system, there may be counted on that the scheduling policy is fair. Consequently, although the implementation is more error prone due to the use of synchronization mechanisms, in the re-engineered version of the VSM³, I pursue the *multi-threaded interpreter approach*.

7.2.2 Data Model Definition

Refactoring *VSM*³ required the extension and adaptation of its *Data Model Definition (DMD)* which was indispensable due to the syntactical and lexical adjustments of *SFML* described in Section 7.1. The *DMD* mainly consists of *AST* classes representing the entities of *SFSCs*, *SFGL*, and *SFSL*. Their in-depth redefinition was needed as preparation of re-engineering the *IRE* and *IDE* of *VSM*³. This also entailed modifications of their *XML* representations used to persistently store the models which were captured by the redesign of the corresponding *XML* schema definition as *XSD*.

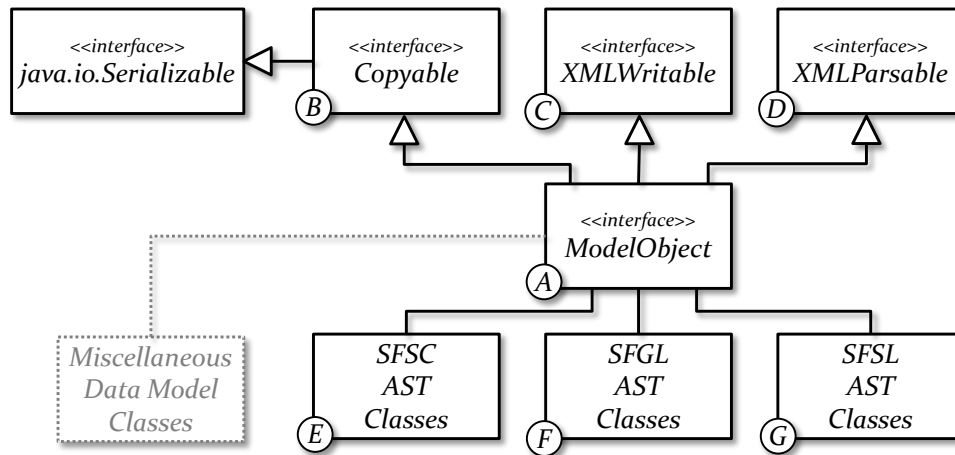


Figure 7.2.3: A diagram showing an extract of the data model definition on the data model layer.

As shown in Figure 7.2.3, the classes of the *DMD* constitute a hierarchy with a *model object interface* as root (Figure 7.2.3 (A)). It extends interfaces declaring functions for *deep copying* via *serialization* (Figure 7.2.3 (B)), *XML stream writing* (Figure 7.2.3 (C)), and *XML parsing* (Figure 7.2.3 (D)) that are implemented by all model classes. Besides some project, configuration and graphics classes, the most important parts of the data model are the *AST* classes of *SFSCs* (Figure 7.2.3 (E)), *SFGL* (Figure 7.2.3 (F)), and *SFSL* (Figure 7.2.3 (G)). The individual *AST* class hierarchies and their implementation straightforwardly ensues the respective syntactical and lexical specifications described in Section 7.1. Figure 7.2.4 shows the hierarchy of that data model's node (Figure 7.2.4 (A)) and edge classes (Figure 7.2.4 (B)) which are part of the *AST* for *SFSCs*. Figure 7.2.5 shows an extract of the class hierarchy of the *SFSL*'s *AST* and Figure 7.2.6 shows important command and expressions classes of *SFGL*'s *AST* and their relationships.

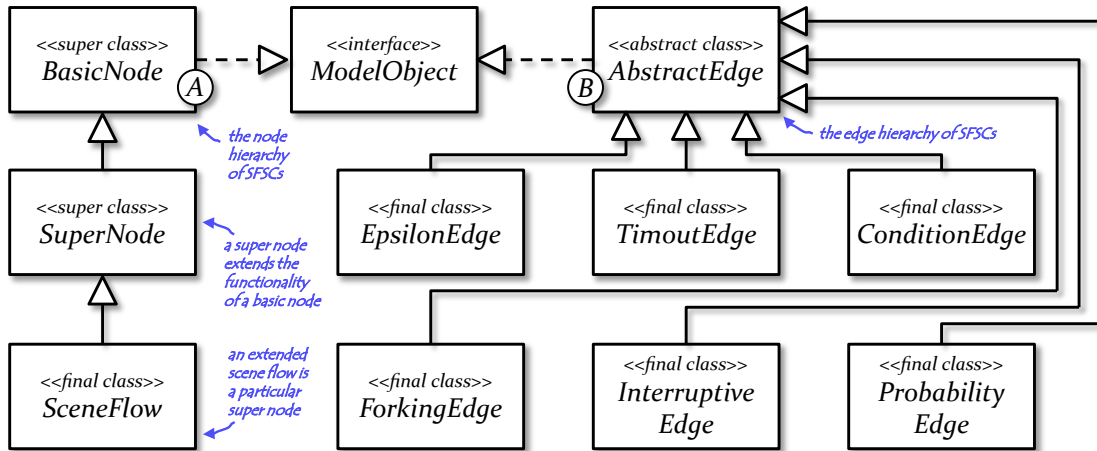


Figure 7.2.4: A diagram partly showing the hierarchy of classes representing the SFSC constructs.

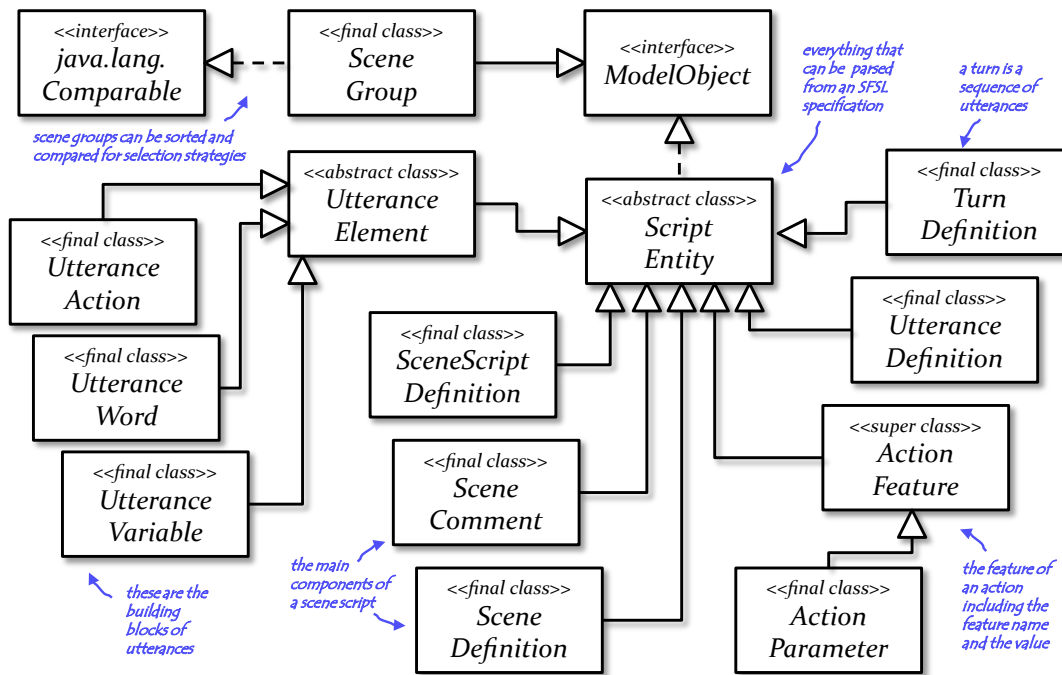


Figure 7.2.5: A diagram partly showing the hierarchy of classes representing the SFSL elements.

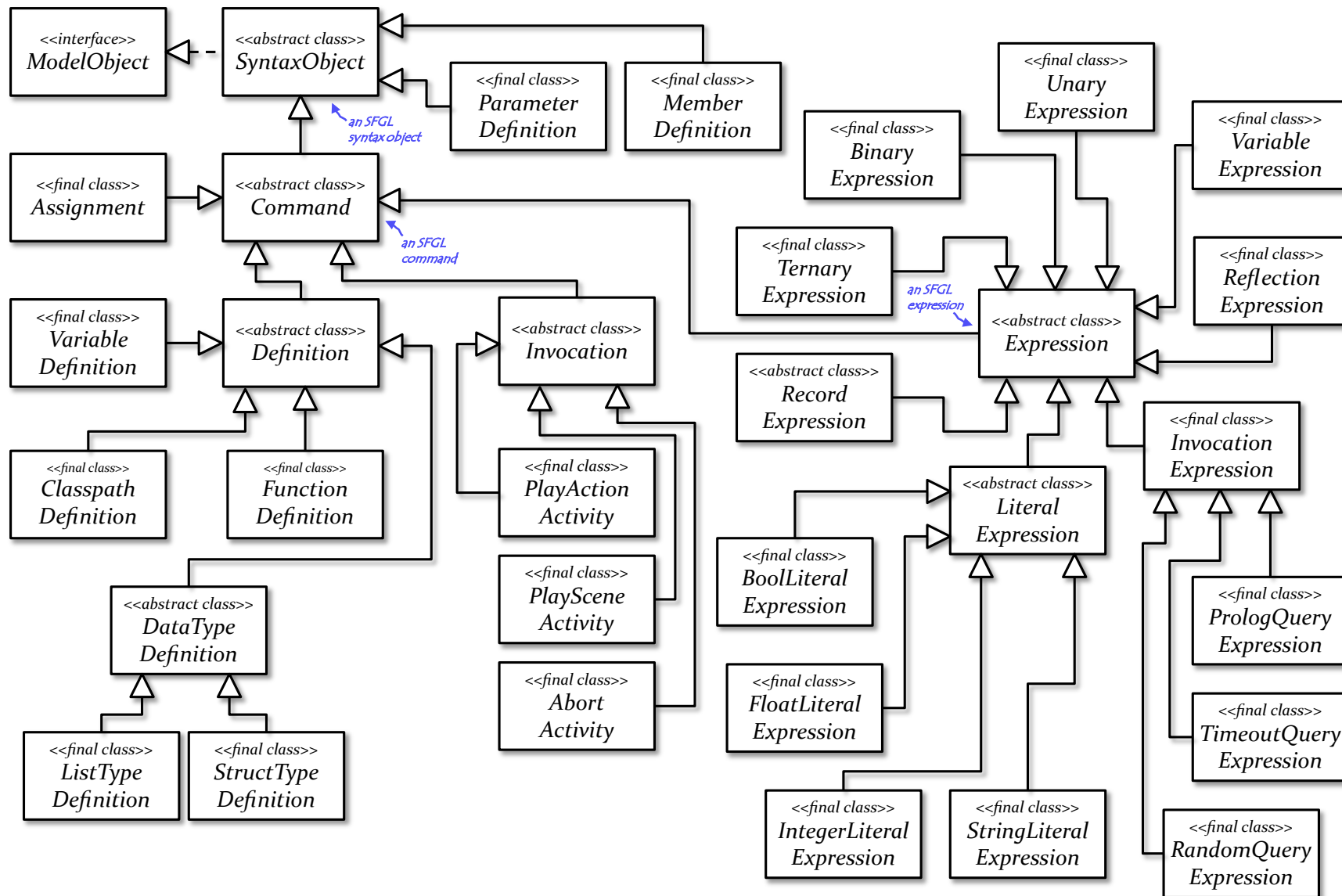


Figure 7.2.6: A diagram partly showing the hierarchy of classes representing the SFSC constructs.

7.2.3 Runtime Environment

The *IRE* on the runtime layer of VSM³ was re-factored based on the *DMD* redefinition. The components of the *IRE* are with their interplay responsible for the execution of *SFSCs*, the evaluation of *SFGL* statements and expressions, the scheduling of behavioral activities specified in *SFSL*, the evaluation of logic queries formulated in *SFQL*, and the real-time transmission of system status information events to the *IDE*.

Core Components

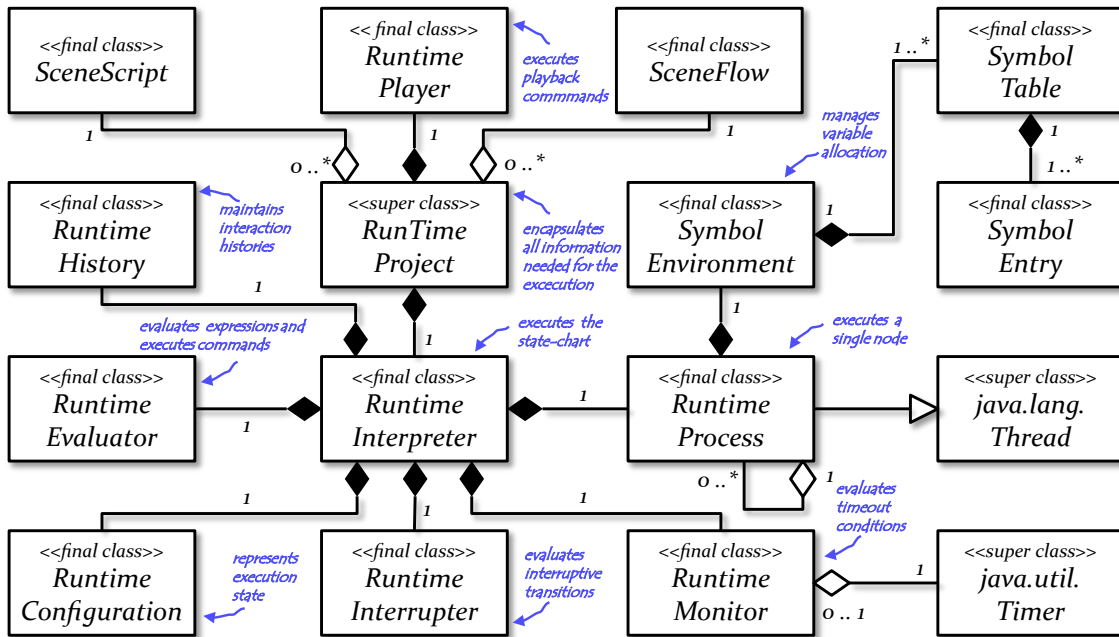


Figure 7.2.7: A diagram showing some important classes of the interpreter runtime environment.

Figure 7.2.7 shows the core classes of the *IRE* with their aggregation and composition relationships. A *runtime project* is encapsulating all model parts and configurations and provides the functions to manage them in a project directory. It comprises a *runtime player* which is responsible for the execution of action and scene playback commands and the proper scheduling of the respective activities. A *runtime interpreter* is executing extended scene flow state-charts using a hierarchy of parallel *runtime processes*. It encapsulate the execution state information of a model, such as, a *runtime configuration* and a *runtime history*. A *runtime evaluator* is used for processing definitions, executing commands, and evaluating expressions. It applies appropriate synchronization mechanisms when a runtime process requires exclusive access to a critical section or shared memory. It is supported by a *runtime monitor* for monitoring and evaluating timeout conditions and a *runtime interrupter* for examining interruptive transitions and eventually initiate their interruption or termination by sending appropriate *runtime signals* to the executing runtime processes and their children in each execution step. A runtime process is a special thread which one after another executes nodes and edges of an extended scene flow within a *symbol environment*. Such an environ-

ment contains a list of *symbol tables*, one for the currently executed node and others for its parent super nodes. A symbol table is simply a list of *symbol entries* each of which is mapping a variable name to a *runtime value*, which can be a primitive, struct, or list value.

Activity Playback

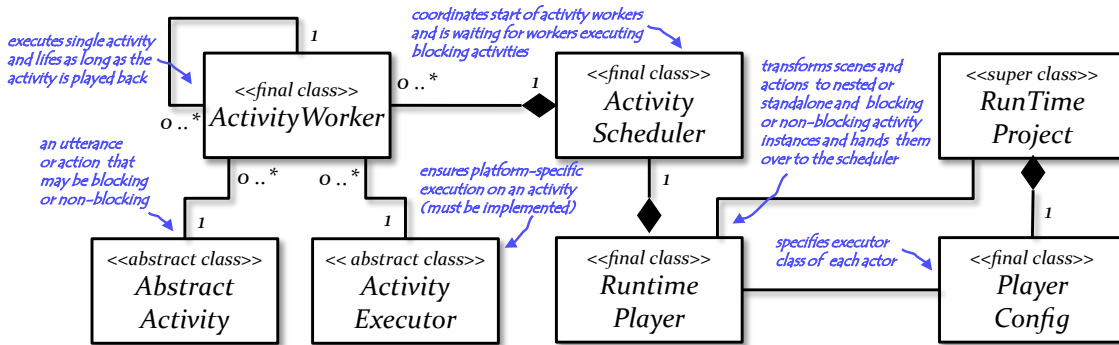


Figure 7.2.8: A diagram showing the relationships between the classes used for activity playback.

Figure 7.2.8 shows the relationships and associations between the runtime player and its correlative components that are together responsible for executing playback commands for scene, action and utterance activities. It is initialized with a project-specific *player configuration* which contains application- and platform-specific information about the individual agents and their characteristics as well as the class paths to the *activity executor* instance plug-ins that must be loaded for the communication with the respective agent-platforms. When playing a scene or action, then the player constructs the corresponding *action activity* and *utterance activity* objects and delegates their scheduling to the *activity scheduler* which relies on *activity worker threads* for the execution of activities on specific executors.

Listing 7.2.1: The execution of a scene using the *playScene* function of the class *RuntimePlayer*.

```

1 public final class RuntimePlayer {
2     // ...
3     public final void playScene(final String name, final String lang, final
4         LinkedList args) {
5         // ...
6         final SceneObject scene =
7             mProject.getSceneScript().getSceneGroup(name, lang).select();
8         // Create playback task
9         final PlayerWorker worker = new PlayerWorker(task) {
10             @Override
11             public void run() {
12                 for (SceneTurn turn : scene.getTurnList()) {
13                     // Get the executor for this turn
14                     final ActivityExecutor turnExecutor =
15                         mProject.getAgentDevice(turn.getSpeaker());
16                     // Serially play the utterances
17                     for (SceneUttr uttr : turn.getUttrList()) {
18                         final LinkedList<String> textBuilder = new LinkedList();
19                         final LinkedList<ActivityWorker> workerList = new LinkedList();
20                         for (final UttrElement element : uttr.getWordList()) {
21                             if (element instanceof ActionObject) {
22                                 final ActionObject action = (ActionObject) element;
23                                 // Get the executor for this action
24                                 final ActivityExecutor actionExecutor =
25                                     (action.getActor() != null ?
  
```

```

25         mProject.getAgentDevice(actor) : turnExecutor);
26         // Create a new marker for the action
27         final String marker = turnExecutor.marker(newId());
28         // Append the marker to the activity
29         textBuilder.add(marker);
30         // Register the activity with marker
31         workerList.add(
32             mScheduler.register(marker, // Execute at marker
33                 new ActionActivity(
34                     (action.getActor() == null) ?
35                         turn.getSpeaker() : action.getActor(),
36                     action.getName(),
37                     action.getText(substitutions),
38                     action.getFeatureList(),
39                     substitutions),
40                     actionExecutor));
41         }
42         else {
43             // Append the text to the activity
44             textBuilder.add(element.getText(substitutions));
45         }
46     }
47     final String punctuation = uttr.getPunctuationMark();
48     // Schedule the utterance activity
49     mScheduler.schedule(
50         0, // Schedule without delay
51         workerList,
52         new SpeechActivity(
53             turn.getSpeaker(),
54             textBuilder,
55             punctuation),
56         turnExecutor);
57     // Check for interruption
58     if (isDone()) {
59         return;
60     }
61 }
62 }
63 }
64 };
65 // Start the playback task
66 worker.start();
67 // Wait for playback task
68 boolean finished = false;
69 while (!finished) {
70     try {
71         // Join the playback task
72         worker.join();
73         // And terminate playback
74         finished = true;
75     } catch (final InterruptedException exc) {
76         // Terminate playback task
77         worker.abort();
78     }
79 }
80 }
81 }

```

Listing 7.2.1 shows an extract of the player's member function which is responsible for the composition and scheduling of activities when playing back a scene. An internal player worker thread is executing the scene and waits until the scene is regularly finished or the calling runtime process, which is waiting for the player worker, is interrupted by the runtime interpreter. After a scene has been selected, the player worker iteratively executes the turns and utterances of this scene one after the other. For each utterance, it creates a new utterance activity and schedules it with zero delay and in blocking mode. For each nested action, it creates an action activity and registers it with a marker at the activity scheduler before it adds

an activity worker for this activity to the list of workers of the utterance activity. The worker executing the utterance activity must wait for the workers executing its nested activities.

Query Execution

Figure 7.2.9 shows the integration of *SWI-PROLOG* using the *JPL 7*⁶ API that can be used to call *JAVA*[™] from *SWI-PROLOG* and vice-versa. The API is wrapped by a *JPL engine* that includes a *JPL loader* to consult the *SWI-PROLOG* modules provided with the runtime project. A *JPL result* class represents the result of a query to *SWI-PROLOG* and extends a list of *JPL terms*, like, for example, atoms, numbers, variables, and compounds terms.

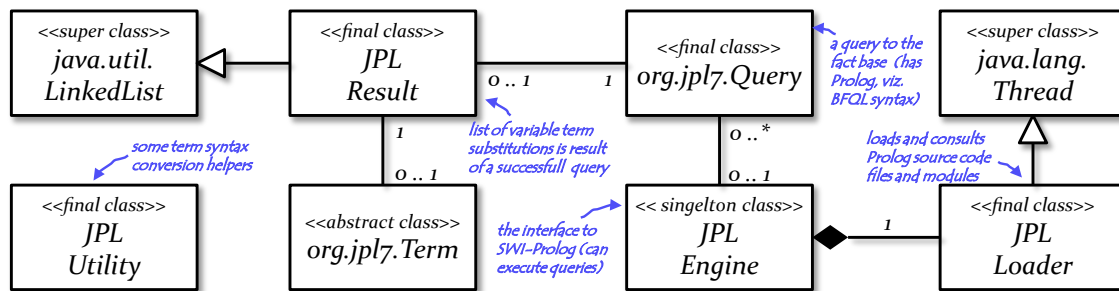


Figure 7.2.9: A diagram showing the relationships between the classes used for query execution.

Listing 7.2.2: The execution of a query using the *query* member function of the class *JPLEngine*.

```

1  import java.util.Map;
2  import org.jpl7.JPL;
3  import org.jpl7.Query;
4  import org.jpl7.Term;
5
6  public final class JPLEngine {
7      // ...
8
9      public final synchronized JPLResult query(String querystr) {
10         // Eventually initialize JPL
11         init();
12         // Create a query for this call
13         final Query query = new Query(querystr);
14         // Create a result for this call
15         final JPLResult result = new JPLResult(query);
16         try {
17             // Get all solutions of the query
18             final Map<String, Term>[] solutions
19                 = query.allSolutions();
20             // Add the solutions to the result
21             for (int i = 0; i < solutions.length; i++) {
22                 result.add(solutions[i]);
23             }
24         } catch (final Exception exc) {
25             sLogger.failure(exc.toString());
26         } finally {
27             query.close();
28         }
29         // Return the result of the query
30         return result;
31     }
32 }

```

⁶<http://jpl7.org/>

Listings 7.2.2 and 7.2.3 show how a *SFGL* query expression is executed in the evaluator and the engine. The query expression takes a string argument representing the *SWI-PROLOG* query. The *SWI-PROLOG* inference engine then finds substitutions for the uninstantiated variables in this query and returns the respective *JPL* result. The evaluator makes a lookup in the environment of the executing *IRE* process and assigns the string representations of the terms that have been unified with the free variables to the corresponding *BFSC* variables.

Listing 7.2.3: The execution of a query using the *query* member function of the class *Evaluator*.

```

1 import java.util.Map;
2
3 public final class Evaluator {
4     // ...
5     public final boolean query(final String querystr, final Environment env) {
6         // Delegate query execution to JPL
7         final JPLResult result = mEngine.query(querystr)
8         // Check the result of the query
9         if (result.size() == 1) {
10            // Get the first and single solution
11            final Map<String, Term> subst = result.getFirst();
12            // Update variables in the environment
13            for (final Entry<String, Term> entry : subst.entrySet()) {
14                try {
15                    final String variable = entry.getKey();
16                    final String binding = JPLUtility.convert(
17                        entry.getValue().toString());
18                    // Throw exception if no such variable
19                    env.write(variable, new StringValue(binding));
20                } catch (Exception exc) {
21                    mLogger.failure(exc.toString());
22                }
23            }
24            return true;
25        } else {
26            return false;
27        }
28    }
29 }

```

7.2.4 Modeling Environment

The third substantial part of *VSM³*, that has been re-factored in this thesis, is the modeling layer which comprises the components of the framework's *IDE*. The *IDE* enables authors to create, maintain, configure, debug, and execute a *VSM³* project with the help of a *graphical user interface*. The *IDE* and *IRE* of *VSM³* can be used *independently* of each other, that means, a project may be executed by the *IRE* in *terminal modus* without instantiating the graphical user interface. This can, for example, be used to save processing power if there is no need for modifications and visualizations at runtime. When used together, both communicate via the central *event dispatching mechanism* of *VSM³*. Since they are simultaneously operating on the project's data model, the interpreter approach allows the modification of model in the *VSM³* and the immediate observation of the ensuing effects at runtime.

Graphical User Interface

Figure 7.2.11 shows a screenshot of the *IDE*'s graphical user interface including the authoring suite's most important editor and configuration components. Figure 7.2.10 shows a diagram

depicting the relationships between the most important classes implementing these components. It can be seen that the *IDE* has been reorganized to a hierarchical architecture in order to implement the update and visualization mechanism using a recursive visitor pattern.

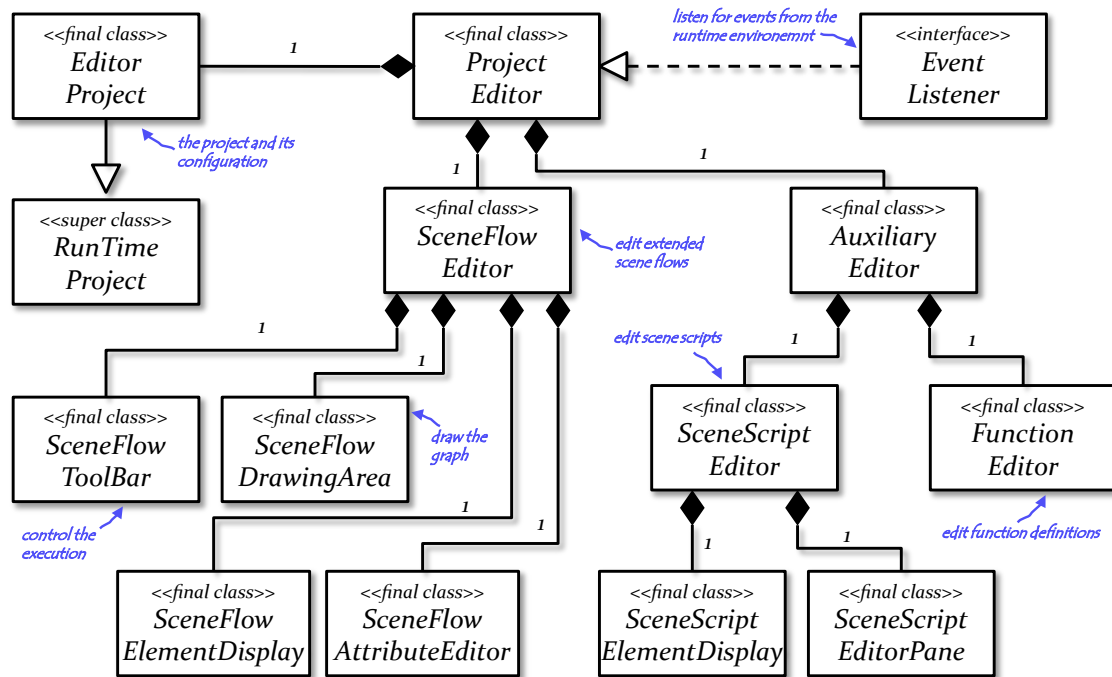


Figure 7.2.10: A diagram showing important classes of the integrated development environment.

The root of this hierarchy is the *project editor* which holds an editor project and implements the event listener interface to register for system events. A central component is the *scene flow editor* (Figure 7.2.11 (A)) which is divided into a *tool bar* (Figure 7.2.11 (B)) and three additional subcomponents. Among those is the *drawing area* (Figure 7.2.11 (C)) in the middle of the *IDE*. It contains the visual representation of the extended scene flow and is used to highlight nodes and edges when they are executed at runtime. To draw a model, an author can drag building blocks from the *element display* (Figure 7.2.11 (D)) on the left and drop them on the drawing area. These blocks can be nodes, edges, and comment badges as well as scenes and predefined *JAVA™* functions. The individual nodes and edges can be edited using the *element editor* (Figure 7.2.11 (E)) on the right side. It allows editing the nodes' names, start nodes, type- and variable definitions, and command statements as well as the properties of the different types of edges when they are selected in the drawing area. Another important part of the *IDE* is the *scene script editor* (Figure 7.2.11 (F)) which is divided into two subcomponents. First, the *editor pane* (Figure 7.2.11 (G)) is used to edit the textual specification of the project's scene script. It supports syntax highlight scene script elements and is also used to highlight scenes when they are executed during runtime. Second, a scene script *element display* (Figure 7.2.11 (H)) contains the available building blocks for scene scripts, such as gestures, animations, and system actions from an optional *gesticon* or *acticon*. The third part of the workspace, which is not shown in the screenshot, is the function editor (Figure 7.2.11 (I)) which allows defining aliases that refer to *JAVA™* functions in the class path that are called via reflection.

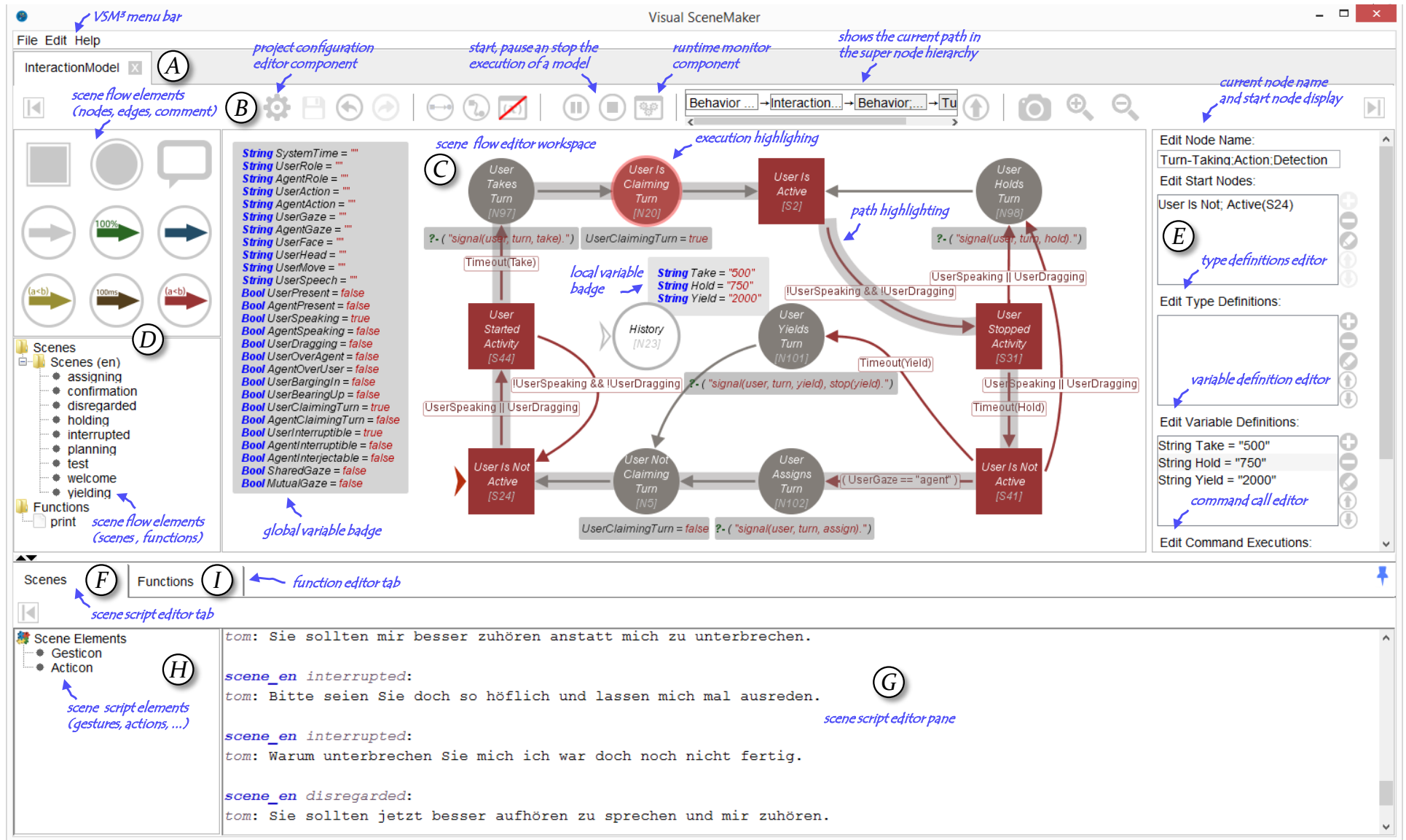


Figure 7.2.11: A screenshot of the IDE of VSM³ showing the authoring suite's most important editor components.

Runtime Visualization

As mentioned before, some components of the *IDE* implement a runtime visualization mechanism that is supposed to facilitate testing, debugging of a model, and verifying its correctness during the modeling phase and, thus, help to assess and control the modeling progress. The mechanism provides a *trace mode* which persistently highlights paths that have been taken within the extended scene flow with a grayish background in order to retrace them after the execution. In addition, it provides a *real-time mode* that highlights currently executed nodes and command statements, edges that are taken and scenes that are played back at runtime with a reddish background. The visualization is implemented with the help of *VSM³*'s central event dispatcher component managing the framework's global event pool. As shown in Figure 7.2.10, the project editor is registered as event listener and propagates visualization events produced by the *IRE* down the component hierarchy of the *IDE* such that an event recursively visits each component of the graphical user interface. Depending on the type of event, each component may decide to either react to the event or ignore it before or after forwarding it to its child components. For example, when a node has been entered, then the executing runtime process generates an event which is only consumed by the node objects currently displayed in the drawing area. The one which refers to the executed node in the data model is then highlighted for some moments using a *highlighting timer thread*. Analogously, when an edge is taken then the corresponding event is examined by the displayed edge objects only, and the one that is actually executed is then highlighted for a short time. While a scene is played back by the runtime player, corresponding events are consumed by the scene editor pane which highlights the area enclosing the definition of this scene in the text area until the scene is finished. Finally, when the value of a variable is changed then the symbol table produces an event which is then consumed by the local and global variable badges of the currently displayed super node. If the variable is in the scope of this super node or any parental node, then the corresponding variable badge is updated to display the actual value of the variable. The described real-time updating and visualization features on the graphical user interface significantly facilitate testing and debugging for an author.

7.3 Developing Demo Applications

For validation purposes, *VSM³* was used to model the interactive behavior of social robots and virtual characters with different tasks and capabilities in a wide range of demonstrator applications in the context of teaching and research projects, workshops, and field tests. Many of them were used to conduct user studies in which *VSM³* significantly facilitated the development of different conditions for creating varying user experiences as well as logging runtime information for the statistical evaluation of these experiments.

7.3.1 Agents in a Virtual School Yard

The use of learning companions in teaching and training environments (Johnson *et al.*, 2000; Gulz, 2004; D'Mello and Graesser, 2013) can, besides possible negative effects (Rickenberg

and Reeves, 2000), increase the learners' commitment to the learning experience, promote their motivation and self-confidence, help to prevent or overcome negative affective states, and minimize undesirable associations with the learning task, such as frustration, boredom, or fear of failure (van Mulken *et al.*, 1998). Teams of pedagogical agents can help the learners to classify the conveyed knowledge and allow the continuous reinforcement of beliefs (André *et al.*, 2000).



Figure 7.3.1: Some hamster characters playing different pedagogical roles in the *DYNALearn* project.

In the *DYNALearn*⁷ project (Bredeweg *et al.*, 2013), we used VSM³ to develop an interactive learning environment that was used by teachers and learners to transmit, express, examine, and improve their conceptual knowledge about cause-effect relations in ecosystems through the use and joint creation of qualitative reasoning models (Bredeweg *et al.*, 2009). Therefore, we designed, created, and evaluated a cast of cartoonish virtual hamster characters with unique pedagogical roles and personalities, shown in Figure 7.3.1, that together form some kind of virtual school yard and enable learners to interact with the software in an easy, intuitive, unobtrusive as well as motivating and engaging way (André *et al.*, 2000). It was our goal to enable teachers to easily adapt lecture and dialog content as well as the interaction and behavior management for their pedagogical agents using VSM³ in order to meet their respective educational demands (Mehlmann *et al.*, 2010).

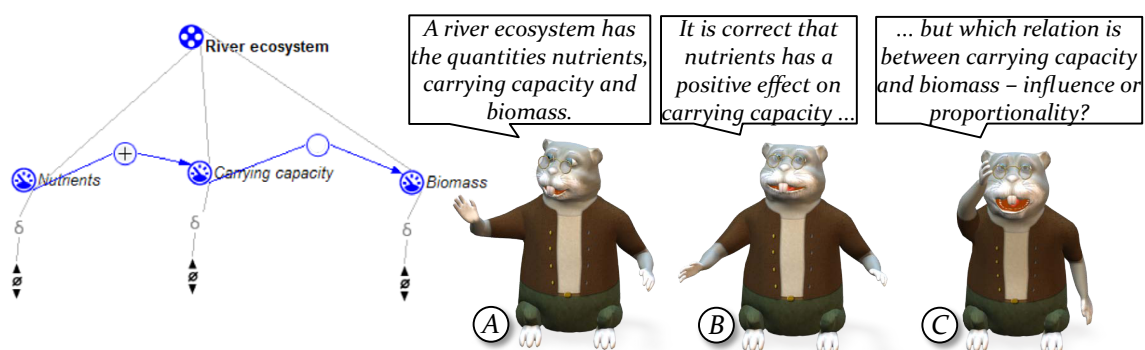


Figure 7.3.2: A use case with the teacher character explaining relations in a conceptual model.

VSM³ has been used to develop a number of use cases that employ different teaching methods, exploit different learning strategies, and use different ways of knowledge conveyance and verification while the learners interact with hamster characters embodying various pedagogical

⁷<https://ivi.fnwi.uva.nl/tcs/QRgroup/DynaLearn/>

ical or educational roles (Wißner *et al.*, 2011, 2012). For example, Figure 7.3.2 shows a lecture and diagnosis phase with an experienced teacher who offers supporting help (Figure 7.3.2 A), feedback (Figure 7.3.2 B), and recommendations (Figure 7.3.2 C) while jointly developing conceptual models with the human learner. Figure 7.3.3 shows the quiz master (Figure 7.3.3 A) in a quiz game with the teachable agent (Figure 7.3.3 B) which has been trained and fed with the user's knowledge before sent to the quiz.

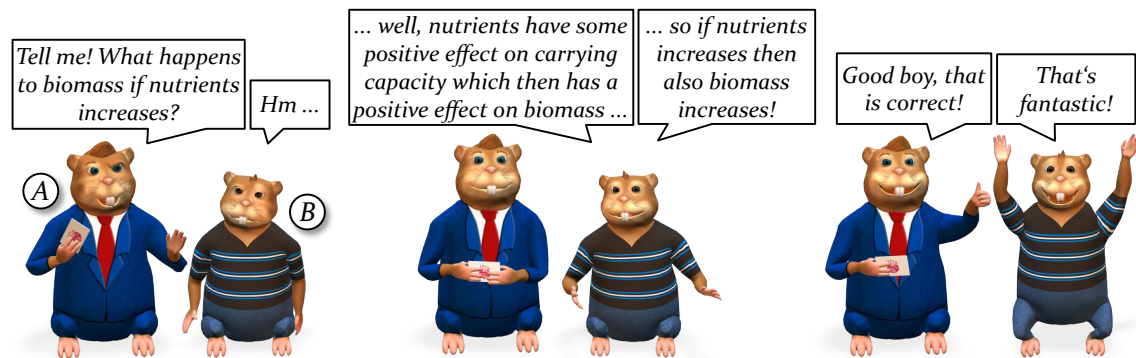


Figure 7.3.3: A use case with the quiz master character playing a quiz with the teachable agent.

Realizing different personalities and roles required, besides the graphical character designs, motion styles, and gesture types, to carefully design the behavior and interaction models to reflect the role-specific behavioral rules and patterns as well as teaching methods and theories from educational science such as *learning by teaching* (Biswas *et al.*, 2009), *scaffolding* (Lipscomb *et al.*, 2001; Larkin, 22), *highlighting* (Cade *et al.*, 2008), and *educational quizzes* (Randel *et al.*, 1992). Therefore, VSM³ was helpful because it allowed the injection of inferred knowledge from the conceptual models into the dialog content for the generation of questions, answers, feedback, and summaries. Furthermore, it enabled the variation of dialog content and nonverbal behavior to avoid wooden and repetitive behavior of the virtual characters. Finally, modeling of the characters in nested parallel processes helped to keep the model clearly arranged and expendable. After a study which aimed solely on the evaluation whether the graphical design communicated the intended roles and personalities (Bühling *et al.*, 2010), a second study evaluated the effect of the characters' functional and dialog behavior, modeled with VSM³. It turned out that the intended suggestion of functionality, roles, and graphical design matched the perception of the studies' participants very well.

7.3.2 Actors in a Virtual Soap Opera

The use of *interactive digital storytelling* (IDS) systems in education and training (Marsella *et al.*, 2000, 2003; Si *et al.*, 2005; Swartout *et al.*, 2006) as well as entertainment and art (Mateas and Stern, 2003b; Riedl *et al.*, 2003; Cavazza *et al.*, 2001) is envisioned to facilitate positive, enjoyable, and moving user experiences (Cavazza *et al.*, 2008; Klimmt *et al.*, 2012). While some of them are putting the user into the role of an observer that can change the world as the story progresses, the majority pursues a *dialog-based* interaction paradigm. They aim at

creating a dramatic experience by offering a selection of conversational situations in which the user can directly influence the progress and outcome of the story.



Figure 7.3.4: The virtual beer-garden environment of the social game *SOAP* in the AAA engine.

In the *IRIS*⁸ project (Cavazza *et al.*, 2008), we used *VSM*³ to develop the interactive narrative-based social game *SOAP* to research different conditions of dialog-based interaction. Figure 7.3.4 shows a screenshot of *SOAP* depicting a virtual beer-garden containing the user's avatar (Figure 7.1.1 A) and several groups of virtual characters (Figure 7.1.1 B, C, D). In the soap-like story of *SOAP*, the user can approach the focus groups, participate in their conversations by typing utterances into a text field (Figure 7.1.1 E), advice the characters and, thus, influencing the progress and outcome of the plot, in the sense of a romantic conflict.

SOAP uses a spell checker and the semantic parser *SPIN* (Engel, 2005) for translating the user's typed-text input into *dialog-acts* (Core and Allen, 1997). Virtual environment and characters, with their automatic low-level behaviors, such as positioning, orientation and proximity are managed with the AAA engine (Damian *et al.*, 2011). A *Bayesian Network* for each agent defines how factors such as personality, emotional state, or culture (Rehm *et al.*, 2007) determine gestural expressiveness parameters, such as speed, extent, or repetition (McNeill, 1992; Pelachaud, 2005) to customize animations in a lexicon. Plot as well as high-level dialog behavior and interaction management are modeled with *VSM*³, such that each focus group, the user avatar, and their behavioral aspects as well as other game objects and input processing processes were modeled in separate, nested and parallel scene flows. Using interruptive transitions and the interaction history, each dialog situation was promptly interruptible when the user leaved a focus group and was consistently reopened and resumed after reentering it in order to create a coherent storytelling experience.

We offered two modes of managing user-character dialogs that come with specific advantages

⁸<http://iris.interactive-storytelling.de/>

and caveats both from a designer's and user's perspective. First, the dialog can be *round-based* (McCoy *et al.*, 2010) forcing the user to become active on predetermined occasions during conversations. Second, it can be *continuous* allowing the users to contribute at any time during the interaction even if this means to interrupt a character. While round-based dialog limits the users' freedom and autonomy to the role of a witness of ongoing conversations, the resulting overall story may be more coherent and the system perceived more comprehensible and usable. In contrast, continuous dialogs can provide an experience that resembles an improvisational theater (Mateas and Stern, 2002) but maintaining a coherent story is a greater challenge, since the system needs to continuously adjust to user input that might be inappropriate or incomprehensible. A comparison of the users' responses to the continuous versus round based mode in *SOAP* showed that users tend to prefer continuous interaction, even though the recognition rate of user utterances was slightly worse than in the round-based version (Endrass *et al.*, 2011). The technologically more ambitious continuous mode was perceived to be closer to film and improvisation theater and seemed to contribute to a more unique, novel kind of user experience while the less demanding round-based mode was judged to be more similar to well-known experiences with classical menu-based adventure games. Finally, this study also showed that *VSM*³ facilitates the conduction and evaluation of experiments of digital storytelling applications in order to help system creators to make better choices when several design options are available (Vermeulen *et al.*, 2010).

7.3.3 Coaches in Interview Trainings

Compared to classical coaching approaches, technology-enhanced training environments can be a viable and advantageous alternative (Sapouna *et al.*, 2010). Their often playful character can increase the learners' enjoyment and motivation while they also create the necessary distance for critical self-reflection. Social skill training in terms of role-playing games with virtual agents, like simulated job interviews, offers great promise to adapt to such socially challenging situations because it provides learners with a realistic, but safe environment that enables them to train particular verbal and nonverbal behaviors that play significant roles during interpersonal interaction (Hollandsworth *et al.*, 1979; Carl, 1980). For example, young unemployed or uneducated adults with low socio-emotional skills (MacDonald, 2008), such as a lack of self-confidence or sense of their own strengths, can practice their stress coping and emotion regulation management to improve their persuasiveness in job interviews.

In the *TARDIS* (Anderson *et al.*, 2013) and *EMPAT* (Langer *et al.*, 2016) projects, we used *VSM*³ to develop *SOCIALCOACH*, a scenario-based serious game simulation platform that supports social application training and coaching by providing different forms of simulated job interviews. Figure 7.3.5 shows some applications of *SOCIALCOACH* (Damian *et al.*, 2013; Baur *et al.*, 2013a; Damian *et al.*, 2015) with virtual *CHARAMEL*⁹ characters in a *TRICAT*¹⁰ environment. *SSI* is used to recognize the user's voice activity, spoken keywords, and social cues for

⁹<http://www.charamel.com/>

¹⁰<https://www.tricat.net/>

emotion regulation and VSM³ is used to model the agents' dialog and interaction behavior.



Figure 7.3.5: The interaction scenario of the *SOCIALCOACH* application in different settings.

During the recruiting process the user is confronted to different interview partners with varying personalities and interview strategies, such as, for example, a more *understanding* (Figure 7.3.6 A) and a rather *demanding* (Figure 7.3.6 B) recruiter. In a consecutive debriefing phase, assisting characters are used to recap, discuss and assess the user's behavior in specific interview situations and to advise them on how to improve their performance. We also developed more *playful* variants in which the user must perform particular behavioral patterns, such as smiling or leaning forward to the interviewer (Figure 7.3.5 A, Figure 7.3.6 C), or must interrupt the agent at specific points during an utterance (Figure 7.3.5 B).



Figure 7.3.6: Some of the virtual characters that the user meets during the job application training.

We used the aforementioned applications to investigate different emotion regulation and coping strategies of users as well as the function of interruptions and the perception of the agent's interruption handling strategies during this kind of social interaction. A first user study with pupils demonstrated clear benefits of the experience-based learning approach with our application over traditional learning methods and showed that the virtual character helped the pupils to better control negative emotional states, such as nervousness (Damian *et al.*, 2015). In a second study, the analysis of the participants' social cues, such as audio features and body language, as well as their subjective judgments confirmed that they felt a higher amount of stress when interacting with a demanding character (Gebhard *et al.*, 2014).

With regard to user interruptions, we wanted to explore to which extent different interruption handling strategies of the agent influence the assessment and perception of the agent's dominance, involvement, and friendliness as well as the comfortableness of the user. A study (Gebhard *et al.*, 2017) revealed that users assess the agent as less dominant, more friendly, and closer when the agent's interruption handling time is short. Moreover, we found that users feel more comfortable to interrupt an agent that stops speaking immediately after the user started talking.

7.3.4 Helpers in Shared Workspaces

Participants of a human interaction constantly establish, maintain, and repair the common ground (Clark, 1996) to avoid or repair disruptions due to misunderstandings, missing attention, or misjudged sensory, perceptive, or cognitive abilities. Gaze is involved in a variety of processes for the generation and recognition of multi-modal and multi-directional behavioral patterns used to reciprocally ensure grounding. Gaze cues are aligned with other modalities to ground the speaker and listener roles (Nielsen, 1962; Duncan, 1972; Kendon, 1967; Sacks *et al.*, 1974), to continually produce, elicit and detect feedback signals (Yngve, 1970; Bavelas *et al.*, 2002), to follow and direct the partners' focus of visual attention to objects or themselves (Argyle *et al.*, 1973; Argyle and Cook, 1976), and to disambiguate verbal references with the speaker's gaze direction (Oviatt, 2003; Oviatt *et al.*, 2015; Staudte and Crocker, 2011). Embedding and coordinating these manifold roles of gaze with each other and the dialog management in a computational behavior and interaction is a complex task.

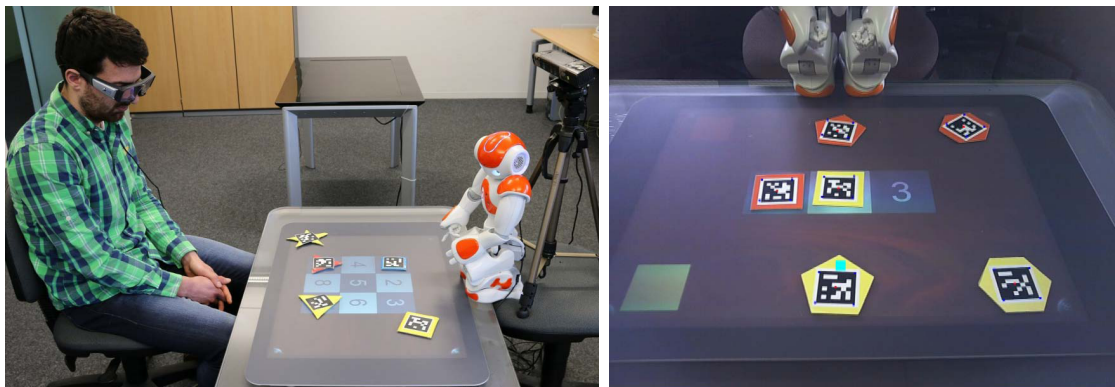


Figure 7.3.7: The interaction scenario of the *ROBOTPUZZLE* application on the shared workspace.

In the *ROBOTPUZZLE* application (Mehlmann *et al.*, 2014b,a, 2016), shown in Figure 7.3.7, we used *VSM*³ to developed such a model for a social *NAO*¹¹ robot that collaborates with the user in a sorting task on a shared *Microsoft*[®]¹² surface table workspace. The user wears *SMI*¹³ eye-tracking glasses and a microphone for speech recognition whose data are fed to an *SSI* pipeline to interpret the user's gaze movements and fixations to objects and areas on the table as well as the user's dialog acts parsed from clarification questions. The robot is supposed

¹¹<http://www.aldebaran-robotics.com/>

¹²<http://www.microsoft.com/>

¹³<http://www.smivision.com/>

to facilitate a sorting task by instructing the user to move puzzle pieces into certain puzzle slots. The pieces have distinguishable features such as a shape, size, color and position and are marked on both sides, to track their position and to recognize the pieces the user is looking at.

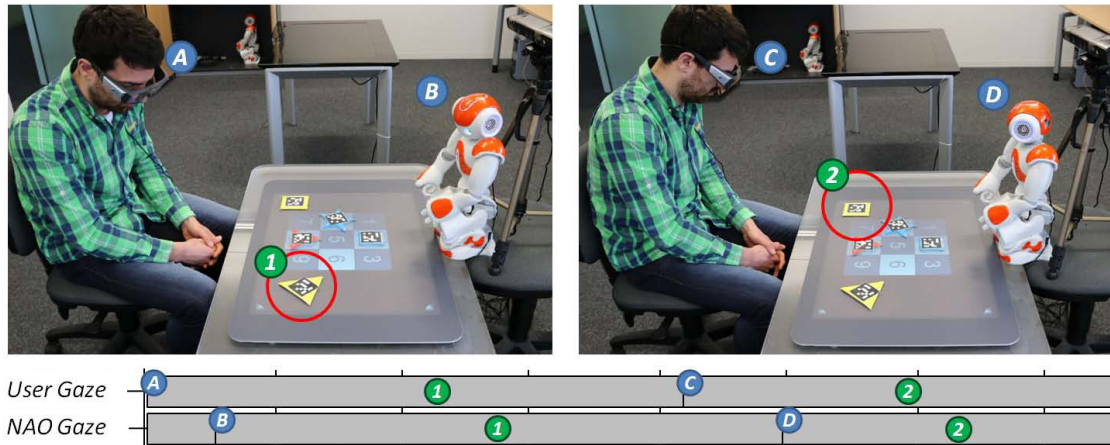


Figure 7.3.8: Robot and user in the *ROBOTPUZZLE* are constantly following each other's attention.

Both interaction partners may use any combination of gaze, pointing gestures and speech to multi-modally refer to the objects, to regulate the speaker and listener roles and to draw the other participant's attention to the objects or themselves. In this, they may produce ambiguous references that can then be resolved by multi-modal disambiguation, combining gaze and speech, or by a clarification dialog. For example, Figure 7.3.8 shows a scene in which the agent is constantly following the user's gaze in order to share his perceptual ground while Figure 7.3.9 shows an example in which the robot takes the user's gaze direction into account to resolve an ambiguous deictic reference in the user's question.

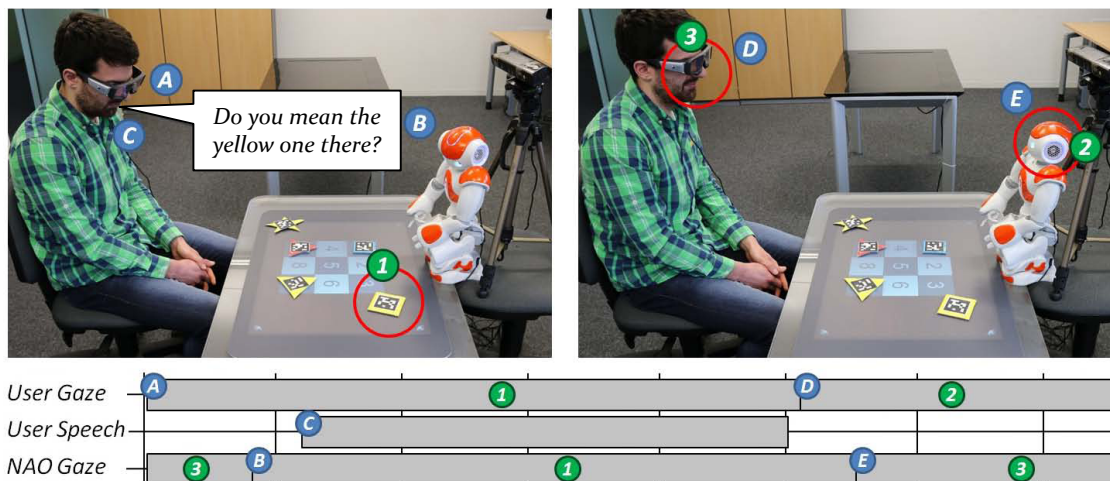


Figure 7.3.9: Robot and user in the *ROBOTPUZZLE* are disambiguating verbal reference with gaze.

VSM³ was helpful because the hierarchical and parallel decomposition allowed modeling the behavioral functions using parallel behavioral and computational processes on different ab-

straction levels and its event logic calculus with temporal constraints and generalized quantification eased the multi-modal disambiguation. In experiments with the applications we investigated the role of the aforementioned gaze functions for the interpersonal coordination and grounding, the effectiveness of the common task, and the social perception of the robot partner (Mehlmann *et al.*, 2014b,a, 2016). Our studies showed that the implemented gaze mechanisms for visual attention sharing and speech disambiguation enable fluent, efficient and pleasant interactions, thus demonstrating the potential of the behavior model in view of interpersonal coordination and grounding.

7.3.5 Further Applications with *VSM*³

During the course of this thesis, many other interesting applications were developed and evaluated with *VSM*³ in order to validate the authoring framework and the underlying modeling approach within the scope of research and teaching projects, workshops, and field tests. Due to the limited space, these are only covered superficially and not explained in detail here.

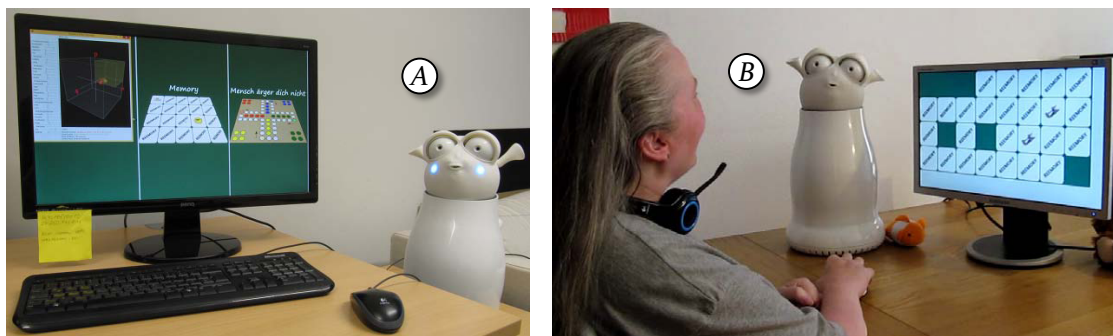


Figure 7.3.10: Applications in which the *Robopec* robot *Reeti* was used as game-playing companion.

Figure 7.3.10 shows our *ROBOTGAMES* application with a social, game-playing robot similar to the relational agent described in (Behrooz *et al.*, 2014). A *Robopec*¹⁴ *Reeti* robot plays different card or board games with the user, for example, to train the mental fitness of elderly people living on their own (Figure 7.3.10 (A)). Our focus here was on the conversational engagement mechanisms and the display of the robot’s cognitive and emotional states. These mechanisms include, for example, mutual gaze with the user, comments about unusual delays, and thinking behavior, such as examining the game screen for options or looking up while remembering the location of a matching card. Furthermore, the robot responds emotionally to various events in the game, both by facial expressions and appropriate comments (Figure 7.3.10 (B)). Ongoing student projects will replace the manually authored emotions with a more sophisticated affect model (Gebhard, 2005; Bee *et al.*, 2010a) to enable autonomous reactions based on the robot’s given personality. We are thus expecting to make the robot’s behavior more credible and to sustain the user’s interest over a long period of time. Others will extend the game playing capabilities of the robot with joke and storytelling functionali-

¹⁴<http://www.reeti.fr/>

ties that can be adapted to individual user personalities using reinforcement learning on the user’s continuously provided social signals (Ritschel and André, 2017).

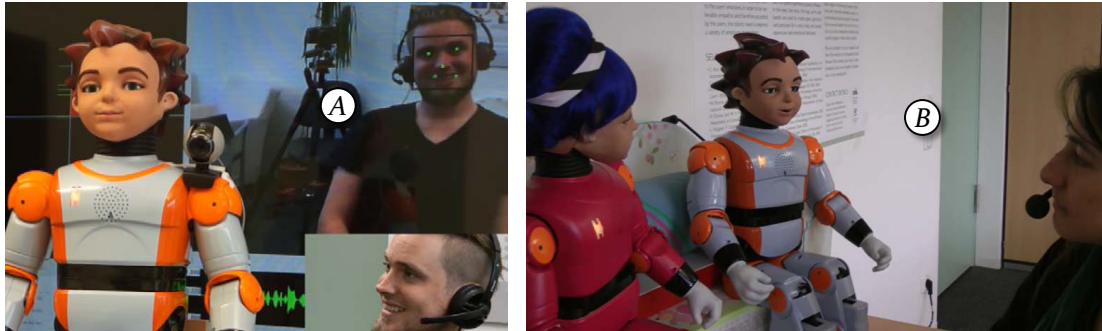


Figure 7.3.11: Applications which use the *RoboKind* robots *Zeno* and *Alice* empathic partners.

Figure 7.3.11 shows photos taken from applications with *RoboKind*¹⁵ robots that we used as test bed for empathy modeling and in which the user’s mood is inferred from various cues such as their tone of voice and facial expression using *SSI* (Figure 7.3.11 A). While the robots’ dialog and interaction flow is mainly modeled with *VSM*³ they concurrently exhibit two distinct empathy mechanisms (Bee *et al.*, 2010a) based on the inferred user emotion (Figure 7.3.11 B). First, they constantly adapt their facial expressions to match the user’s emotion, signaling a basic awareness of their situation. Second, they verbally express happiness or pity for the user depending on his emotional display. Unlike the direct and seemingly instinctive mirroring, this reflects an active interest in the user’s well-being, a key requirement for a social companion. We are planning to evaluate this technology and its use in various future interactive applications, such as personalized recommender systems (Hammer *et al.*, 2016), to allow higher reasoning and constructive advice. For example, the robot might suggest a meeting with friends when the user complains about loneliness or offer to call a doctor when the user is feeling sick. We expect the empathy display to provide additional comfort and encourage the user to take their companion’s advice.

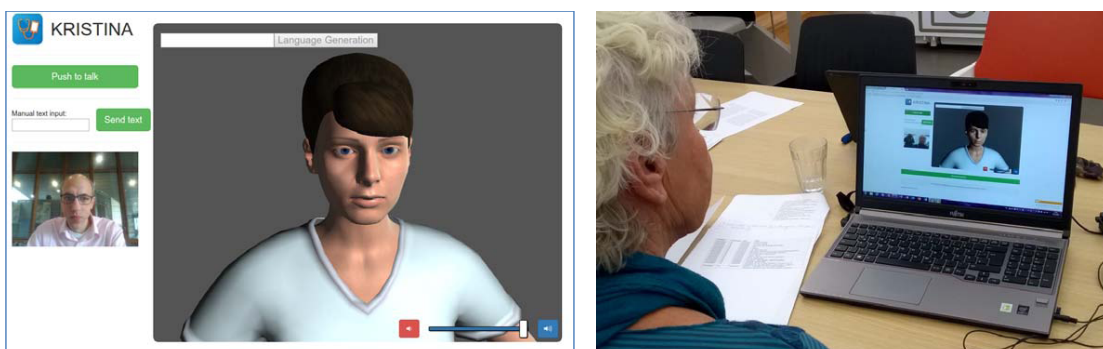


Figure 7.3.12: An application in which the *KRISTINA* agent interacts with elderly and migrants.

¹⁵<http://www.robokindrobots.com/>

Figure 7.3.12 shows an application from the *KRISTINA* project¹⁶ in which an intelligent embodied conversational agent with linguistic, cultural, social, and emotional competence interacts with elderly and migrants in different basic care and health-care scenarios (Wanner *et al.*, 2016). *VSM*³ is used as central coordinating instance managing the data and control flow in a complex distributed software architecture including, among others, the reasoning-driven dialog manager *OWLSPEAK* (Ultes and Minker, 2014) and *SSI* (Wagner *et al.*, 2013) for user input interpretation. Besides this role as central data switchboard, *VSM*³ is in control of the agent's turn-taking behavior arranging the agent's participant role changes based on the observed user input and the agent's own contributions planned by the dialog management. Turn-taking decisions are made on the basis of a policy which determines whether the agent is allowed to interrupt the user's utterance and how it reacts to the user's attempts to barge in its own turn. Finally, it controls a variety of appropriate and vivid nonverbal idle behavior patterns, for example, mimicking the user's facial expressions, gestures, or body postures to make an impression of engagement and attentiveness during the different roles and when these are not negotiated.



Figure 7.3.13: Some photos taken during field tests in the context of nationwide promotion programs.

Figure 7.3.13 shows some photos taken during several field tests that were conducted in the context of different nationwide promotion programs and similar events (Endrass *et al.*, 2010) in which middle school students used *VSM*³ to create interactive performances and social games with several of the aforementioned virtual characters and robots, such as *Reeti* (Figure 7.3.13 (A)), the *HamsterLab* (Figure 7.3.13 (B)) as well as *Zeno* and *Alice* (Figure 7.3.13 (C)). After a short introduction to the tool, the input devices and agent platforms, they were grouped in teams of a few students, they were given about one hour for brainstorming, sketching the interaction and dialog flow, writing scripts, and formulating input constraints. Afterwards, they modeled the scenarios with minimal assistance in about another hour and the frequently very remarkable results were viewed by the whole group. Finally, the students were asked to fill out an evaluation sheet in which they generally gave very positive feedback about the authoring experience with *VSM*³. In addition to the many field tests, we also conducted a usability study in which the tool reached excellent usability scores (Gebhard *et al.*, 2012). We found that the users generally felt very confident with the tool and that the visual pro-

¹⁶<http://kristina-project.eu/>

gramming approach and the provided modeling concepts are easily comprehensible and let non-experts easily create social agent applications in a rapid prototype fashion.

7.4 Summary and Conclusion

This chapter presented the realization of the modeling approach proposed in Chapter 5 in the authoring software *VSM³*. The modeling language ensemble defined in the conceptual part of this thesis has been implemented by the adaptation and extension of the existing set of modeling languages in this authoring framework. *VSM³* uses a multi-threaded interpreter runtime environment for the execution of behavior and interaction models. This allows the extension and modification of a model and the direct observation of the effects without the need for an intermediate code generation step. *VSM³* comes with an integrated development environment which allows visually modeling behavior and interaction models using a graphical editor. The graphical modeling environment allows the runtime visualization of a model's execution. All these features significantly facilitate testing and debugging a social agent's behavior and interaction model. Thus, the authoring suite encourages and supports authors with different background, experience, and expertise to exploit their knowledge in creating behavior and interaction models for social agents. For validation purposes, *VSM³* has been used for the development of various applications with embodied conversational characters and social robot companions. In this, the tool has been used by various user groups, ranging from completely unexperienced teenage girls, over screenwriters and social psychologists, to highly experienced experts.

CHAPTER 8

CONCLUSION — SUMMARY, CONTRIBUTIONS AND FUTURE WORK

In Chapter 1, I explained my motivation for this thesis and highlighted its scientific significance. I introduced an exemplary human-agent interaction scenario based on which I identified and illustrated my research objectives. In Chapter 2, I provided a profound review of literature from social and behavioral sciences to introduce the theoretical background on interpersonal coordination and grounding as well as the functions of gaze behaviors and voice overlaps that contribute to them. In Chapter 3, I discussed the key modeling tasks and requirements that an author is faced with when modeling a social agents interactive behavior with a focus on interpersonal coordination and grounding. In Chapter 4, I showed that, despite substantial research effort, related work has not yet managed to develop modeling frameworks that allow integrating and coordinating the behavioral aspects of interpersonal coordination and grounding in a social agent's behavior and interaction model. Instead, related research either focused on individual behavioral functions in isolation only, or, developed solutions for specific modeling tasks without considering the entire picture. In Chapter 5, I presented the conceptual framework of the modeling approach proposed in this thesis. It allows an author to successfully master the coordination of the behavioral functions and aspects that underlie interpersonal coordination and grounding. It has a remarkable practicability while being sufficiently expressive to go beyond related state-of-the-art efforts. In Chapter 6, I illustrated the approach based on a sophisticated behavior and interaction model which integrates the functions of gaze and voice for interpersonal coordination and grounding. This model can serve as best practice for other authors that want to craft their own behavior and interaction models with the proposed approach. Finally, in Chapter 7, I explained how the approach was implemented in the *VSM*³ authoring software and presented applications that have been developed to validate the tool and underlying the approach.

In the reminder of this chapter, in Section 8.1, I briefly summarize the main methodical, conceptual, and technical contributions of this dissertation with regard to the research objectives introduced in Section 1.3. Finally, in Section 8.2, I outline future development opportunities and research directives that I consider as interesting and promising after the scientific exchange and experiences made in the course of this dissertation.

8.1 Contributions

The overall goal of this dissertation was the design of a modeling framework that allows modeling the integration and coordination of the behavioral functions and processes that underlie interpersonal coordination and grounding in social interactions. Throughout this thesis, it has become clear that this is a fundamental human capability that is, for this reason, also crucial for artificially and socially intelligent agents in order to show credible, engaging, and natural social behavior. Any discrepancies in the synchronization or prioritization of behavioral functions or processes entail the danger these agents appear unnatural, incompetent, or clumsy to their human interaction partners. This thesis has made several methodical, conceptual, and technical contributions on the way to tackling this research objective.

8.1.1 Scientific Approach and Footing

This thesis is the very first scientific attempt to focus on interpersonal coordination and grounding and their synergistic connection for a holistic modeling approach to the interactive behavior of social agents. A methodical contribution of this thesis is therefore the systematic elaboration of the possible interlinking of these two interactional phenomena with the functions of different gaze behaviors and speech overlaps in social joint activities. Therefore, Chapter 2 contains an exhaustive literature survey that explains the various behavioral functions of gaze and voice behaviors, such as attention following, multi-modal disambiguation, turn management, intimacy regulation, feedback eliciting, multi-modal disambiguation, and interrupt handling. Chapter 3 then uses the illustrative example scenario from Chapter 1 to discuss the characteristics of social interactions and physically situated, joint activities and comprehensively carves out how these individual behavioral functions and their underlying behavioral processes contribute to interpersonal coordination and grounding. This theoretical and illustrative groundwork is then taken as a basis to categorize and formulate the modeling tasks and requirements as well as the solution concepts for the modeling approach proposed in this thesis.

8.1.2 The *BFML* Modeling Framework

MODELING APPROACH DESIGN The, for certain, most important conceptual contribution of this thesis is the design of the modeling approach with *BFML* that allows coordinating the behavioral functions and processes that contribute to interpersonal coordination and grounding. In the conceptual design of this approach, I argue for dividing the creation of a social agent's behavior and interaction model into three modeling subtasks. Then, I identify three task-specific requirements that a modeling approach must meet in order to enable an author to master these subtasks. Finally, I present the design of an ensemble of domain-specific modeling languages each of which is successfully tackling one of these subtasks. In this, it shows that the modeling framework proposed in this thesis is the first to combine the benefits of a specially designed, hierarchical and concurrent state-chart variant, a domain-specific, logic calculus, and a template-based behavior specification language for modeling interactive behavior of artificially and socially

intelligent agents. This remarkably practicable and expressive modeling framework successfully masters the research goals of this thesis.

Coordinating Functions & Processes

The domain-specific state-chart variant *BFSC* is used to control the interplay of behavioral functions and processes contributing to interpersonal coordination and grounding. It allows the *incremental and reciprocal meshing* of input processing, knowledge reasoning and behavior generation. It enables the *parallel and hierarchical structuring* of a model through its *parallel decomposition* and *hierarchical refinement* into parallel and nested, behavioral and computational processes on different behavioral levels. It allows the immediate *interruption and coherent resumption* of behavioral functions and processes in reaction to quickly changing behavioral goals and priorities.

Integrating Input & Context Events

The *PROLOG*-embedded, domain-specific, logic calculus *BFQL* is used for multi-modal fusion and knowledge reasoning. It uses a *uniform representation format* for input events that are maintained in a *well-organized working memory* to preserve their chronological order. Logic predicates are used for *multi-modal fusion and reasoning* based on semantic, temporal, and quantitative integration constraints and dynamic predicates are used to manage a garbage collection mechanism on the event history.

Creating Behavior & Dialog Content

The template-based description format *BFSL* is used for the specification of expressive and natural behavioral activities. It allows *versatile compositions of behavior* resembling the wide range of human behavioral and linguistic repertoire. It allows the *flexible integration of knowledge* to create competent and informed behavior and dialog content. It supports the *automatic variability of behavior* to avoid repetitions that have a negative impact on an agent's plausibility.

The examination of advantages and limitations of previous modeling approaches and a comparison to our own solution in Chapter 4 shows several points. First, with respect to expressiveness, the proposed novel modeling framework goes way beyond the state-of-the-art related approaches because it successfully masters the complex coordination and interplay of the many behavioral functions that underlie interpersonal coordination and grounding. The approach is by this sufficiently usable since it uses mostly declarative and visual modeling paradigms as well as uniform representation formats.

An important part of the conceptual contribution in this thesis is the development of a sophisticated, exemplary behavior and interaction model in Chapter 6, using the proposed modeling approach. Therefore, Chapter 6 follows to a very great extent a top-down, that means theory-driven approach, oriented along literature from behavioral psychology and related work on human-agent interaction. In some cases, the models are adapted according to observations made in an analysis of human-agent interaction corpora by related work. The

MODELING
APPROACH
ILLUSTRATION

step-wise developed, highly modular model is designed to be generic, adaptable, reusable, and comprises a large part of the roles of gaze behaviors and voice overlaps that contribute to interpersonal coordination and grounding. The model can serve as best practice example and toolbox resource of individual reusable and adaptable parts for authors that want to craft their own behavior and interaction models. In this, the modeling approach architectures as well as the behavior and interaction models of related work, presented and discussed in Chapter 4, can be recreated and combined and their results can be reproduced using the modeling framework in this thesis.

8.1.3 The *VSM*³ Authoring Software

The main technical contribution of this thesis is the reference implementation of the novel modeling framework in the authoring software *VISUALSCENEMAKER*³ (*VSM*³). This tool has been developed in order to encourage, guide, and support authors in the distributed and iterative development of interactive applications with social companions. Part of the technical contribution is the validation of the *VSM*³ authoring suite, and the underlying modeling approach, in a number of representative, fully fledged applications. In this, *VSM*³ has successfully been utilized for the development of interactive embodied conversational agents and social robots in a variety of teaching projects for the use as an educational tool, in a wide range of research projects with different requirements and objectives, and field tests. In this, the tool has successfully been used by different user groups, such as computer experts, artists, screenwriters, social psychologists, and even teenagers. This is certainly a clear sign for the outstanding expressive power but also the adaptability and reusability of the modeling framework for authors with different background knowledge, expertise, experience, and modeling strategies. With the help of *VSM*³, it is now a realistically attainable objective that these authors can exploit their expert knowledge in the respective areas for the development of computational behavior and interaction models of social companions.

8.2 Future Work

During the work on this thesis, the *VSM*³ software has been used by various user groups in different application areas. This manifold use has led to a great deal of constructive feedback how to extend the modeling formalisms and authoring framework in order to improve the created behavior and interaction models. Of course, when working with people that have different interests and expertises, it becomes inevitable and sometimes difficult to accommodate their differing aims for the future development. However, a number of achievable extensions to the work in this thesis have become most apparent. This includes different future research directives, concerning methodical improvements and conceptual extensions, as well as the associated further development prospects.

8.2.1 Corpus-Based Model Refinement

The illustrative behavior flows in Chapter 6 have been modeled following, for the most part, a top-down, that means theory-driven approach. The choice of most parameters has been oriented along literature from social and behavioral sciences. Theories from social and behavioral sciences, that are based on the analysis of human interactions, are a good guideline and starting point for the creation of theoretically well-grounded computational models. However, they are often rather vague when it comes to the details and parameterization of a model, such as, for example, the exact values of timeouts or distributions of probabilities. For that reason, whenever available, but unfortunately only in few cases, specific parameter values have been obtained from observations made in corpus-driven analyses of related work on human-agent interaction. However, even the values reported in this literature are not infrequently heavily wide apart or even contradictory.

To avoid these issues, it would be useful to record, annotate, and analyze multi-modal corpora recordings of the interactions in the applications that have been developed with the approach proposed in this thesis. This would help to obtain more precise and informative statistical data and to iteratively adapt the parameters of a model until a satisfactory result has been achieved. It would also be interesting to investigate in how far it is possible to even automatically learn and generate, and thus avoid hand-crafting, structural details of behavior flows based on the observation of real user interaction data. This could establish an iterative development cycle in which the results from the evaluation studies and annotations are used as input to refine the behavior and interaction models. A first step has already been taken with the development of a plug-in for the annotation tool *ANVIL*¹ (Kipp, 2001, 2014). This plug-in allows the simulation of the input for the model by taking the annotations, transforming them into events and feeding the model as if these events would be produced by the participants or the environment in real-time. This approach significantly improves the examination of the model for parts that are worthy of improving, for example, timing constraints of timeout edges or the priorities of behavioral aspects.

*OBSERVATION-
BASED MODEL
REFINEMENT*

8.2.2 Data-Driven Behavior Adaptation

An interesting further development and research directive is the enrichment of behavior flows with data-driven prediction models that have been created using machine learning methods. Just like the models in Chapter 6, behavior flows usually expose different *neuralgic points* at which decisions are made, timeouts are awaited, or behavioral responses are determined. Instead of exclusively relying on the current rule-based approach, it might help to produce more intelligent behaviors when falling back on assessments by a more specialized and sophisticated mechanism that is based on a data-driven prediction model.

For example, the model in Chapter 6 is, in its current form, only able to detect interruption attempts after it has already detected a voice or turn overlap of a specified fixed length. However, humans reveal particular nonverbal and para-verbal behaviors, such as breathing

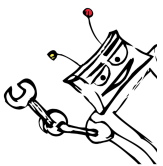
*INTEGRATING
PREDICTION
MODELS*

¹<http://www.anvil-software.org>

patterns, gaze cues, facial expressions, and gestures that signal an imminent attempt to grab the floor or change the topic. The meaning of these signals for the turn-taking decisions depends on additional parameters, such as the interpersonal relationship and the partner's personality traits. A data-driven prediction model could be trained with such features and then be able to anticipate an interruption attempt even before the user starts speaking. The behavior flow could continuously request this information in order to make sure that the agent proactively and appropriately responds to these predictions. Vice-versa, another prediction model can assesses if a situation is opportune for the agent to attempt to grab the floor itself. Based on context information and social cues, the model would predict the probability for the occurrence of transition-relevant points in the conversation and the likelihood that an attempt to take the turn will be accepted by the user.

**LEARNING
ADAPTIVE
MODELS** Another possibility to improve the models created with the modeling approach in this thesis would be to make them automatically adaptive to user-specific needs, their personalities, or interpersonal attitudes and relationships to the agent. Such influencing factors could be initially captured by a dedicated user model (Gebhard, 2005). Based on such a model's assessments, the agent could, for example, decide if, how intense and how long to mimic an user's facial expression in order to signal empathy or not. Furthermore, it could also infer the best time when to avert the gaze again in order to balance their interpersonal intimacy. Then, it would be interesting to refine such models using *reinforcement learning* approaches that allow steadily learning specific model parameters from the interaction with the users. For example, (Ritschel and André, 2017) already managed to learn some kind of user humor model that describes the user's joke and storytelling preferences. Similar, the agent could learn, for example, to use the user's continuously provided social signals, such as gaze aversion or emotional displays, to learn the degree of interpersonal intimacy that the user perceives as most adequate and comfortable. Therefore, it could, for example, systematically vary and adapt the length and intensity of its attempts to mirror the user's emotional displays to balance the intimacy regulation whenever the user searches for mutual gaze.

THE END I am convinced that, in the near future, an ensemble of reasonably combined and closely coordinated, both theory-grounded and data-driven models, is the most promising method to make progress with the behavior and interaction models of artificially intelligent social agents. With the modeling approach and authoring framework presented in this thesis, that is particularly suitable for exactly these integrative and coordinative responsibilities, I am confident to have contributed my small part towards achieving this aim.



BIBLIOGRAPHY

- Andrea Abele. Functions of Gaze in Social Interaction: Communication and Monitoring. *Journal of Nonverbal Behavior*, 10(2):83–101, 1986. [Cited on pages 5 and 38.]
- Reginald B. Adams and Robert E. Kleck. Effects of Direct and Averted Gaze on the Perception of Facially Communicated Emotion. *Emotion, American Psychological Association*, 5(1):3–11, 2005. [Cited on page 37.]
- Henny Admoni and Brian Scassellati. Social Eye Gaze in Human-Robot Interaction: A Review. *Journal of Human-Robot Interaction*, 6(1):26–63, 2017. [Cited on page 74.]
- Jan Alexandersson and Tilman Becker. The Formal Foundations Underlying Overlay. In *Proceedings of the 5th International Workshop on Computational Semantics, IWCS 2003, Tilburg, Netherlands, January, 2003*, pages 22–36, 2003. [Cited on pages 91, 94, and 119.]
- James F. Allen and George Ferguson. Actions and Events in Interval Temporal Logic. *Journal of Logic Computation, Oxford University Press, Oxford, UK*, 4(5):531–579, July 1994. [Cited on pages 65 and 127.]
- James F. Allen and Patrick J. Hayes. Moments and Points in an Interval-based Temporal Logic. *Computational Intelligence, Blackwell Publishers, Inc. Cambridge, MA, USA*, 5(4):225–238, November 1990. [Cited on pages 65 and 127.]
- James F. Allen. An Interval-Based Representation of Temporal Knowledge. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence, IJCAI 1981, Vancouver, BC, Canada, August 24–28, 1981*, pages 221–226, 1981. [Cited on pages 65 and 127.]
- James F. Allen. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM, ACM, New York, NY, USA*, 26(11):832–843, November 1983. [Cited on pages 52, 65, and 127.]
- James F. Allen. Towards a General Theory of Action and Time. *Journal of Artificial Intelligence*, 23(2):123–154, July 1984. [Cited on pages 65 and 127.]
- James F. Allen. Rethinking Logics of Action and Time. In *Proceedings of the 2013 20th International Symposium on Temporal Representation and Reasoning, TIME 2013, Pensacola, FL, USA, September 26–28, 2013*, pages 3–4. IEEE Computer Society Washington, DC, USA, 2013. [Cited on pages 65 and 127.]
- Jürgen Allgayer, Karin Harbusch, Alfred Kobsa, Carola Reddig, Norbert. Reithinger, and Dagmar Schmauks. XTRA: A Natural-Language Access System to Expert Systems. *International Journal of Man-Machine Studies*, 31(2):161–195, August 1989. [Cited on page 89.]
- Paul D. Allopenna, James S. Magnuson, and Michael K. Tanenhaus. Tracking the Time Course of Spoken Word Recognition using Eye Movements: Evidence for Continuous Mapping Models. *Journal of Memory and Language*, 38(4):419–439, May 1998. [Cited on page 34.]
- Jens Allwood, Joakim Nivre, and Elisabeth Ahlsén. On the Semantics and Pragmatics of Linguistic Feedback. *Journal of Semantics, also in Gothenburg Papers in Theoretical Linguistics 64, Dept of Linguistics, Göteborg University, 1992*, 9(1):1–26, 1993. [Cited on pages 5, 17, 36, 37, 43, 54, 55, 70, and 179.]
- Keith Anderson, Elisabeth André, Tobias Baur, Sara Bernardini, Mathieu Chollet, Evi Chryssafidou, Ionut Damian, Cathy Ennis, Arjan Egges, Patrick Gebhard, Hazael Jones, Magalie Ochs, Catherine Pelachaud, Kaska Porayska-Pomsta, Paola Rizzo, and Nicolas Sabouret. The TARDIS Framework: Intelligent Virtual Agents for Social Coaching in Job Interviews. In Dennis Reidsma, Haruhiro Katayose, and Anton Nijholt, editors, *Proceedings of the 10th International Conference on Advances in Computer Entertainment, ACE 2013, Boekelo, The Netherlands, November 12–15, 2013*, volume 8253 of *Lecture Notes in Computer Science*, pages 476–491. Springer International Publishing, 2013. [Cited on page 233.]

- Anne H. Anderson. The Effects of Face-to-Face Communication on the Intelligibility of Speech. *Journal of Perception and Psychophysics*, 59(1):580–592, 1997. [Cited on page 34.]
- Elisabeth André and Thomas Rist. Presenting Through Performing: On the Use of Multiple Lifelike Characters in Knowledge-Based Presentation Systems. In *Proceedings of the 2nd International Conference on Intelligent User Interfaces, IUI 2000, New Orleans, LA, USA, January 9-12, 2000*, pages 1–8, 2000. [Cited on page 83.]
- Elisabeth André and Thomas Rist. Controlling the Behavior of Animated Presentation Agents in the Interface: Scripting versus Instructing. *AI Magazine*, 22(5):53–66, 2001. [Cited on page 83.]
- Elisabeth André, Kim Binsted, Kumiko Tanaka-Ishii, Sean Luke, Gerd Herzog, and Thomas Rist. Three RoboCup Simulation League Commentator Systems. *AI Magazine, American Association for Artificial Intelligence*, 21(1):57–66, Spring 2000. [Cited on page 82.]
- Elisabeth André, Thomas Rist, Susanne Van Mulken, Martin Klesen, and Stephan Baldes. The Automated Design of Believable Dialogues for Animated Presentation Teams. In Justine Cassell, Joseph Sullivan, Scott Prevost, and Elizabeth F. Churchill, editors, *Embodied Conversational Agents*. MIT Press Cambridge, MA, USA, 2000. [Cited on page 230.]
- Elisabeth André, Jean-Claude Martin, Florian Lingenfeller, and Johannes Wagner. Multimodal Fusion in Human-Agent Dialogue. In M. Rocj and N. Campbell, editors, *Coverbal Synchrony in Human-Machine Interaction*, pages 387–410. CRC Press, Taylor & Francis Group, 2014. [Cited on pages 10, 63, and 119.]
- Elisabeth André. Challenges for Social Embodiment. In *Proceedings of the 2014 Workshop on Roadmapping the Future of Multimodal Interaction Research including Business Opportunities and Challenges, RFMIR 2014, Istanbul, Turkey, November 12 - 16, 2014*, pages 35–37. ACM New York, NY, USA, 2014. [Cited on pages 63 and 66.]
- Sean Andrist, Xiang Zhi Tan, Michael Gleicher, and Bilge Mutlu. Conversational Gaze Aversion for Humanlike Robots. In *Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction*, pages 25–32, 2014. [Cited on pages 39, 78, and 99.]
- Darko Anicic, Paul Fodor, Sebastian Rudolph, Roland Stühmer, Nenad Stojanovic, and Rudi Studer. A Rule-Based Language for Complex Event Processing and Reasoning. In Pascal Hitzler and Thomas Lukasiewicz, editors, *Proceedings of the 4th International Conference on Web Reasoning and Rule Systems, RR 2010, Bressanone/Brixen, Italy, September 22-24, 2010*, volume 6333 of *Lecture Notes in Computer Science*, pages 42–57. Springer Berlin Heidelberg, 2010. [Cited on page 118.]
- Darko Anicic, Paul Fodor, Sebastian Rudolph, and Nenad Stojanovic. EP-SPARQL: A Unified Language for Event Processing and Stream Reasoning. In *Proceedings of the 20th international Conference on World Wide Web - Session: Query and Ontology Languages, WWW 2011, Hyderabad, India, March 28-April 1, 2011*, pages 635–644. ACM New York, NY, USA, 2011. [Cited on page 118.]
- Michael Argyle and Mark Cook. *Gaze and Mutual Gaze*. Cambridge University Press, Cambridge, UK, January 1976. [Cited on pages 5, 27, 30, 34, 37, 68, 171, 193, and 235.]
- Michael Argyle and Janet Dean. Eye-Contact, Distance and Affiliation. *Sociometry*, 28(3):289–304, September 1965. [Cited on pages 5, 31, 38, and 39.]
- Michael Argyle and JeanAnn Graham. The Central Europe Experiment: Looking at Persons and Looking at Objects. *Environmental Psychology and Nonverbal Behavior, Kluwer Academic Publishers-Human Sciences Press*, 1(1):6–16, 1976. [Cited on pages 32, 34, and 193.]
- Michael Argyle and Roger Ingham. Gaze, Mutual Gaze, and Proximity. *Semiotica - Journal of the International Association for Semiotic Studies*, 6(1):32–49, January 1972. [Cited on page 193.]
- Michael Argyle, Florisse Alkema, and Robin Gilmour. The Communication of Friendly and Hostile Attitudes by Verbal and Non-Verbal Signals. *European Journal of Social Psychology*, 1(3):385–402, July/September 1971. [Cited on page 38.]
- Michael Argyle, Roger Ingham, Florisse Alkema, and Margaret McCallin. The Different Functions of Gaze. *Semiotica, De Gruyter*, 7(1):19–32, 1973. [Cited on pages 15, 34, 193, and 235.]
- Michael Argyle. Non-Verbal Communication in Human Social Interaction. In Robert A. Hinde, editor, *Non-Verbal Communication*. Cambridge University Press, Oxford, UK, 1972. [Cited on page 56.]
- Michael Argyle. *Bodily Communication*. Methuen and Co Ltd, London, 1975. [Cited on page 56.]

- Hassan Ait-Kaci, Andreas Podelski, and Gert Smolka. A Feature Constraint System for Logic Programming with Entailment. *Journal of Theoretical Computer Science, Elsevier Science Publishing, New York, NY, USA*, 122(1-2):263–283, January 1994. [Cited on page 119.]
- Ruth Aylett, Joao Dias, and Ana Paiva. An Affectively Driven Planner for Synthetic Characters. In *Proceedings of the International Conference on Automated Planning and Scheduling, ICAPS 2006, The English Lake District, Cumbria, UK, June 6-10, 2006*, pages 2–10. AAAI press, 2006. [Cited on page 84.]
- Rolf Backofen and Gert Smolka. A Complete and Recursive Feature Theory. *Journal of Theoretical Computer Science, Elsevier Science Publishing, New York, NY, USA*, 146(1-2):243–268, July 1995. [Cited on page 119.]
- J. W. Backus, F. L. Bauer, J. Green, C. Katz, J. McCarthy, P. Naur, A. J. Perlis, H. Rutishauser, K. Samelson, B. Vauquois, J. H. Wegstein, A. van Wijngaarden, and M. Woodger. Revised Report on the Algorithmic Language ALGOL 60. *Computer Journal*, 5(4):349–349, 1963. [Cited on page 109.]
- Norman I. Badler, Bonnie L. Webber, Welton Becket, Christopher W. Geib, Michael B. Moore, Catherine Pelachaud, Barry D. Reich, and Matthew Stone. Planning and Parallel Transition Networks: Animation's New Frontiers. In S. Y. Shin and T. L. Kunii, editors, *Postprint Version. Reprinted from Computer Graphics and Applications: Proceedings of Pacific Graphics 1995*, pages 101–117. World Scientific Publishing: River Edge, NJ, USA, 1995. [Cited on page 86.]
- G rard Bailly, Stephan Raidt, and Fr d ric Elisei. Gaze, Conversational Agents and Face-To-Face Communication. *Journal of Speech Communication*, 52(6):598–612, June 2010. [Cited on page 74.]
- Stephan Baldes, Patrick Gebhard, Martin Klesen, Peter Rist, Thomas Rist, and Markus Schmitt. The interactive CrossTalk Installation: Meta-Theater with Animated Presentation Agents. In *Proceedings of the International Workshop on Lifelike Animated Agents held at the 7th Pacific Rim International Conference on Artificial Intelligence, PRICAI 2002, August 19, Tokyo, Japan, 2002*, pages 9–15, 2002. [Cited on pages 85 and 199.]
- Robert F. Bales, Fred L. Strodbeck, Theodore M. Mills, and Mary E. Roseborough. Channels of Communication in Small Groups. *American Sociological Review*, 16(4):461–468, August 1951. [Cited on page 36.]
- Robert F. Bales. *Personality and Interpersonal Behavior*. Holt, Rinehart and Winston, New York, NY, USA, 1970. [Cited on page 36.]
- Olivier Balet. INSCAPE: An Authoring Platform for Interactive Storytelling. In M. Cavazza and S. Donikian, editors, *Proceedings of the 4th International Conference on Virtual Storytelling: Using Virtual Reality Technologies for Storytelling, ICVS 2007, Saint-Malo, France, December 5 - 7, 2007*. Springer-Verlag Berlin, Heidelberg, 2007. [Cited on page 88.]
- Srinivas Bangalore and Michael Johnston. Integrating Multimodal Language Processing With Speech Recognition. In *The Proceedings of the 6th International Conference on Spoken Language Processing, ICLSP 2000, Beijing International Convention Center, Beijing, China, October 16-20, 2000*, pages 1–4, 2000. [Cited on page 94.]
- Srinivas Bangalore and Michael Johnston. Robust Understanding in Multimodal Interfaces. *Journal of Computational Linguistics, Association for Computational Linguistics*, 35(3):345–397, September 2009. [Cited on pages 92, 94, and 118.]
- Adrian Bangerter, Eric Chevalley, and Sylvie Derouwaux. Managing Third-Party Interruptions in Conversations: Effects of Duration and Conversational Role. *Journal of Language and Social Psychology*, 29(2):235–244, June 2010. [Cited on page 60.]
- Adrian Bangerter. Using Pointing and Describing to Achieve Joint Focus of Attention in Dialogue. *Psychological Science, American Psychological Society*, 15(6):415–419, June 2004. [Cited on page 34.]
- Ankica Bariic, Vasco Amaral, and Miguel Goul o. Usability Evaluation of Domain-Specific Languages. In *Proceedings of the 8th International Conference on the Quality of Information and Communications Technology, QUATIC 2012, Lisbon, Portugal, September 3 - 6, 2012*, pages 342–347, 2012. [Cited on page 104.]
- Jim Barnett, Michael Bodell, Daniel C. Burnett, Jerry Carter, and Rafah Hosn. State Chart XML (SCXML): State Machine Notation for Control Abstraction. W3C Working Draft, February 2007. [Cited on page 86.]
- Reuben M. Baron and Louis A. Boudreau. An Ecological Perspective on Integrating Personality and Social Psychology. *Journal of Personality and Social Psychology*, 53(6):1222–1228, December 1987. [Cited on page 20.]

- Simon Baron-Cohen, Sally Wheelwright, Jacqueline Hill, Yogini Raste, and Ian Plumb. The "Reading the Mind in the Eyes" Test Revised Version: A Study with Normal Adults, and Adults with Asperger Syndrome or High-Functioning Autism. *Journal of Child Psychology and Psychiatry*, 42(2):241–251, February 2001. [Cited on pages 5, 33, and 54.]
- Simon Baron-Cohen. The Eye Direction Detector (EDD) and the Shared Attention Mechanism (SAM): Two Cases for Evolutionary Psychology. In C. Moore and P. J. Dunham, editors, *Joint Attention: Its Origins and Role in Development*, pages 41–59. Erlbaum, Hillsdale, NJ, USA, 1995. [Cited on pages 31 and 34.]
- Simon Baron-Cohen. *Mindblindness: An Essay on Autism and Theory of Mind*. MIT Press, Cambridge, Massachusetts, London, England, 1997. [Cited on pages 19 and 37.]
- Jon Barwise and Robin Cooper. Generalized Quantifiers and Natural Language. *Journal of Linguistics and Philosophy*, 4(2):159–219, 1981. [Cited on pages 108 and 130.]
- Tobias Baur, Ionut Damian, Patrick Gebhard, Kaska Porayska-Pomsta, and Elisabeth André. A Job Interview Simulation: Social Cue-Based Interaction with a Virtual Character. In *Proceedings of the 2013 ASE/IEEE International Conference on Social Computing, SocialCom 2013, Washington D.C., USA, September 8-14, 2013*, pages 220–227. IEEE Computer Society Washington, DC, USA, 2013. [Cited on pages 151, 155, and 233.]
- Tobias Baur, Ionut Damian, Florian Lingenfeller, Johannes Wagner, and Elisabeth André. NovA: Automated Analysis of Nonverbal Signals in Social Interactions. In Albert Ali Salah, Hayley Hung, Oya Aran, and Hatice Gunes, editors, *Proceedings of the 4th International Workshop on Human Behavior Understanding, HBU 2013, In Conjunction with ACM Multimedia 2013, Barcelona, Spain, October 22, 2013*, volume 8212 of *Lecture Notes in Computer Science*, pages 160–171. Springer-Verlag New York, Inc. New York, NY, USA, 2013. [Cited on page 154.]
- Tobias Baur, Gregor U. Mehlmann, Ionut Damian, Patrick Gebhard, Florian Lingenfeller, Johannes Wagner, Birgit Lugrin, and Elisabeth André. Context-Aware Automated Analysis and Annotation of Social Human-Agent Interactions. *Special Issue of the ACM Transactions on Interactive Intelligent Systems: Behavior Understanding for Arts and Entertainment, Article No. 11, ACM New York, NY, USA*, 5(2):1–8, July 2015. [Cited on pages 153, 154, and 155.]
- Janet B. Bavelas, Alex Black, Charles R. Lemery, and Jennifer Mullet. "I show how you feel": Motor Mimicry as a Communicative Act. *Journal of Personality and Social Psychology*, 50(2):322–329, February 1986. [Cited on pages 18 and 19.]
- Janet B. Bavelas, Alex Black, Charles R. Lemery, and Jennifer Mullet. Empathy and its Development. In N. Eisenberg and J. Strayer, editors, *Motor Mimicry as Primitive Empathy*, pages 317–338. Cambridge University Press, New York, NY, USA, 1987. [Cited on page 19.]
- Janet B. Bavelas, Linda Coates, and Trudy Johnson. Listener Responses as a Collaborative Process: The Role of Gaze. *Journal of Communication, International Communication Association*, 52(3):566–580, September 2002. [Cited on pages 5, 17, 35, 36, 37, 54, 55, 68, 179, 181, and 235.]
- Geoffrey W. Beattie. The Role of Language Production Processes in the Organization of Behavior in Face-To-Face Interaction. In Brian Butterworth, editor, *Language Production*, pages 69–107. Academic Press, London, 1980. [Cited on pages 35, 37, and 44.]
- Geoffrey W. Beattie. Interruption in Conversational Interaction, and its Relation to the Sex and Status of the Interactants. *Journal of Linguistics*, 19(1-2):15–36, January 1981. [Cited on pages V, 40, 41, 44, 60, and 173.]
- Geoffrey W. Beattie. A further Investigation of the Cognitive Interference Hypothesis of Gaze Patterns during Conversation. *British Journal of Social Psychology*, 20(4):243–248, November 1981. [Cited on page 37.]
- Nikolaus Bee, Elisabeth André, and Susanne Tober. Breaking the Ice in Human-Agent Communication: Eye-Gaze Based Initiation of Contact with an Embodied Conversational Agent. In Zsófia Ruttkay, Michael Kipp, Anton Nijholt, and Hannes Högni Vilhjálmsón, editors, *Proceedings of the 9th International Conference on Intelligent Virtual Agents, IVA 2009, Amsterdam, The Netherlands, September 14-16, 2009*, pages 229–242. ACM, New York, NY, USA, Springer-Verlag, Berlin, Heidelberg, 2009. [Cited on pages 78 and 99.]
- Nikolas Bee, Elisabeth André, Thuriid Vogt, and Patrick Gebhard. Close Engagements with Artificial Companions: Key Social, Psychological, Ethical and Design Issues. In Yorick Wilks, editor, *The Use of Affective and Attentive Cues in an Empathic Computer-Based Companion*, pages 131–142. John Benjamins Publishing Company, 2010. [Cited on pages 37, 237, and 238.]

- Nikolaus Bee, Colin Pollock, Elisabeth André, and Marilyn Walker. Bossy or Wimpy: Expressing Social Dominance by Combining Gaze and Linguistic Behaviors. In Jan M. Allbeck, Norman I. Badler, Timothy W. Bickmore, Catherine Pelachaud, and Alla Safonova, editors, *Proceedings of the 10th International Conference on Intelligent Virtual Agents, IVA 2010, Philadelphia, Pennsylvania, USA, September 20-22, 2010*, volume 6356 of *Lecture Notes in Computer Science*, pages 265–271. Springer-Verlag, Berlin, Heidelberg, 2010. [Cited on pages 78, 99, and 193.]
- Nikolaus Bee, Johannes Wagner, Elisabeth André, Thurid Vogt, Fred Charles, David Pizzi, and Marc Cavazza. Discovering Eye Gaze Behavior During Human-agent Conversation in an Interactive Storytelling Application. In Wen Gao, Chin-Hui Lee, Jie Yang, Xilin Chen, Maxine Eskénazi, and Zhengyou Zhang, editors, *Proceedings of the International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction, ICMI-MLMI 2010, Beijing, China, November 8-12, 2010*, pages 1–8, ACM, New York, NY, USA, 2010. [Cited on pages 78 and 99.]
- Morteza Behrooz, Charles Rich, and Candace Sidner. On the Sociability of a Game-Playing Agent: A Software Framework and Empirical Study. In Timothy Bickmore, Stacy Marsella, and Candace Sidner, editors, *Proceedings of the 14th International Conference on Intelligent Virtual Agents, IVA 2014, Boston, MA, USA, August 27-29, 2014*, volume 8637 of *Lecture Notes in Computer Science*, pages 40–53. Springer International Publishing, Switzerland, 2014. [Cited on pages 6 and 237.]
- Yacine Bellik. Media Integration in Multimodal Interfaces. In *Proceedings of the IEEE 1st Workshop on Multimedia Signal Processing, Princeton, New Jersey, USA*, pages 31–36, 1997. [Cited on page 64.]
- Yacine Bellik. Technical Requirements for a Successful Multimodal Interaction. In *International Workshop on Information Presentation and Natural Multimodal Dialogue, IPNMD 2001, Verona, Italy, December 14–15, 2001*, 2001. [Cited on page 64.]
- Adrian Bennett. Interruptions and the Interpretation of Conversation. *Journal of Discourse Processes*, 4(2):171–188, April 1981. [Cited on pages 5, 15, 39, 42, 43, and 56.]
- Christian Benoit, Jean-Claude Martin, Catherine Pelachaud, Lambert Schomaker, and Bernhard Suhm. Audio-Visual and Multimodal Speech Systems. *Handbook of Standards and Resources for Spoken Language Systems-Supplement*, 500:1–95, 2000. [Cited on page 89.]
- Frank J. Bernieri and Robert Rosenthal. Interpersonal Coordination: Behavior Matching and Interactional Synchrony. In Robert Stephen Feldman and Bernard Rimé, editors, *Fundamentals of Nonverbal Behavior. Studies in Emotion and Social Interaction*, pages 401–432. Cambridge University Press, Paris, France, Editions de la Maison des Sciences de l'Homme, 1991. [Cited on pages 5, 15, 16, 17, 18, 19, 20, 21, 42, 59, 60, and 68.]
- Frank J. Bernieri, Janet M. Davis, Robert Rosenthal, and C. Raymond Knee. Interactional Synchrony and Rapport: Measuring Synchrony in Displays Devoid of Sound and Facial Affect. *Personality and Social Psychology Bulletin*, 20(4):303–311, June 1994. [Cited on pages 21 and 29.]
- Frank J. Bernieri. Coordinated Movement and Rapport in Teacher-Student Interactions. *Journal of Nonverbal Behavior*, 12(2):120–138, June 1988. [Cited on page 21.]
- Douglas Biber. *Variation across Speech and Writing*. Cambridge University Press, 1988. [Cited on page 71.]
- Timothy Bickmore, Daniel Schulman, and George Shaw. DTask and LiteBody: Open Source, Standards-Based Tools for Building Web-Deployed Embodied Conversational Agents. In James L. Crowley, Yuri A. Ivanov, Christopher Richard Wren, Daniel Gatica-Perez, Michael Johnston, and Rainer Stiefelwagen, editors, *Proceedings of the 11th International Conference on Multimodal Interfaces and the Sixth Workshop on Machine Learning for Multimodal Interaction, ICMI-MLMI 2009, Cambridge, Massachusetts, USA, November 2-6, 2009*, pages 425–431. ACM, New York, NY, USA, 2009. [Cited on page 84.]
- Timothy W. Bickmore, Daniel Schulman, and Candace L. Sidner. A Reusable Framework for Health Counseling Dialogue Systems Based on a Behavioral Medicine Ontology. *Journal of Biomedical Informatics*, 44(2):183–197, April 2011. [Cited on page 84.]
- Gautam Biswas, Rod Roscoe, Hogyong Jeong, and Brian Sulcer. Promoting Self-Regulated Learning Skills in Agent-Based Learning Environments. In *Proceedings of the 17th International Conference on Computers in Education, ICCE 2009, Hong Kong, Hong Kong, November 30 - December 4, 2009*, pages 67–74, 2009. [Cited on page 231.]

- Daniel Bobbert and Magdalena Wolska. Dialog OS: An Extensible Platform for Teaching Spoken Dialogue Systems. In Ron Artstein and Laure Vieu, editors, *Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, Decalog 2007, Trento, Italy, 30 May – 1 June 2007*, pages 159–160, 2007. [Cited on page 85.]
- Kathryn Bock and Willem J. M. Levelt. Language Production: Grammatical Encoding. In M. A. Gernsbacher, editor, *Handbook of Psycholinguistics*, pages 945–984. Academic Press, San Diego, CA, USA, 1994. [Cited on page 23.]
- Dan Bohus and Eric Horvitz. Models for multiparty engagement in open-world dialog. In *Proceedings of the SIGDIAL 2009 Meeting on Discourse and Dialogue, SIGDIAL 2009, London, UK, September 11-12, 2009*, pages 225–234, 2009. [Cited on page 170.]
- Dan Bohus and Eric Horvitz. Facilitating Multiparty Dialog with Gaze, Gesture, and Speech. In Wen Gao, Chih-Hui Lee, Jie Yang, Xilin Chen, Maxine Eskénazi, and Zhengyou Zhang, editors, *Proceedings of the International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction, ICMI-MLMI 2010, Beijing, China, November 8-12, 2010*, pages 1–8, ACM, New York, NY, USA, 2010. [Cited on pages 170, 183, and 184.]
- Dan Bohus and Eric Horvitz. On the Challenges and Opportunities of Physically Situated Dialog. In *Dialog with Robots, Papers from the 2010 AAAI Fall Symposium, Arlington, Virginia, USA, November 11-13, 2010. AAAI Technical Report FS-10-05, AAAI 2010*, pages 2–7, 2010. [Cited on page 5.]
- Dan Bohus and Eric Horvitz. Technical report MSR-TR 2010-115. Technical report, Microsoft Research, 2010. [Cited on pages 77, 170, and 171.]
- Dan Bohus and Eric Horvitz. Multiparty Turn Taking in Situated Dialog: Study, Lessons, and Directions. In *Proceedings of the SIGDIAL 2011 Meeting on Discourse and Dialogue, SIGDIAL 2011, Portland, Oregon, June 17-18, 2011*, pages 98–109, 2011. [Cited on pages 170 and 180.]
- Dan Bohus and Alexander I. Rudnicky. Ravenclaw: Dialog Management using Hierarchical Task Decomposition and an Expectation Agenda. In *Proceedings of the 8th European Conference on Speech Communication and Technology, EUROSPEECH 2003 - INTERSPEECH 2003, Geneva, Switzerland, September 1-4, 2003*. ISCA, 2003. [Cited on pages 84, 118, and 190.]
- Dan Bohus and Alexander I. Rudnicky. The RavenClaw Dialog Management Framework - Architecture and Systems. *Computer Speech and Language*, 23(3):332–361, July 2009. [Cited on pages 84 and 99.]
- Richard A. Bolt. Put-That-There : Voice and Gesture at the Graphics Interface. In *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1980, Seattle, Washington, USA, July 14 - 18, 1980*, pages 262–270. ACM, New York, NY, USA, 1980. [Cited on page 89.]
- Wauter Bosma and Elisabeth André. Exploiting Emotions to Disambiguate Dialogue Acts. In *Proceedings of the 9th International Conference on Intelligent User Interfaces, IUI 2004, Funchal, Madeira, Portugal, January 13-16, 2004*, pages 85–92. ACM New York, NY, USA, 2004. [Cited on pages 63 and 66.]
- Jean-David Boucher, Ugo Pattacini, Amelie Lelong, Gerard Bailly, Frederic Elisei, Sascha Fagel, Peter Ford Dominey, and Jocelyne Ventre-Dominey. I Reach Faster When I See You Look: Gaze Effects in Human–Human and Human–Robot Face-to-Face Cooperation. *Journal on Frontiers in Neurobotics*, 6(3), 2012. [Cited on pages 77 and 99.]
- Jullien Bouchet, Laurence Nigay, and Thierry Ganille. ICARE Software Components for Rapidly Developing Multimodal Interfaces. In *Proceedings of the 6th International Conference on Multimodal Interfaces, ICMI 2004, State College, PA, USA, October 13 - 15, 2004*, pages 251–258. ACM, New York, NY, USA, 2004. [Cited on page 90.]
- Marie-Luce Bourguet and Jaeseung Chang. Design and Usability Evaluation of Multimodal Interaction with Finite State Machines: A Conceptual Framework. *Journal on Multimodal User Interfaces*, 2(1):53–60, July 2008. [Cited on pages 95 and 133.]
- Marie-Luce Bourguet. A Toolkit for Creating and Testing Multimodal Interface Designs. In Michel Beaudouin-Lafon, editor, *Posters of the 15th Annual ACM Symposium on User Interface Software and Technology, UIST 2002, Paris, France, October 27-30, 2002*, volume 2, pages 29–30. ACM, New York, NY, USA, 2002. [Cited on page 94.]
- Marie-Luce Bourguet. Designing and Prototyping Multimodal Commands. In Matthias Rauterberg, Marino Menozzi, and Janet Wesson., editors, *Human-Computer Interaction, INTERACT 2003: IFIP TC13 International Conference on Human-Computer Interaction, Zurich, Switzerland, September 1-5, 2003*, pages 1–4, 2003. [Cited on page 94.]

- Marie-Luce Bourguet. How Finite State Machines Can Be Used To build Error Free Multimodal Interaction Systems. In E. O'Neill, P. Palanque, and P. Johnson, editors, *The 17th British HCI Group Annual Conference, Designing for Society, Bath, UK, September 8-12, 2003*. Springer-Verlag, London, UK, 2003. [Cited on page 94.]
- Marie-Luce Bourguet. Software Design and Development of Multimodal Interaction. In René Jacquart, editor, *Building the Information Society, IFIP 18th World Computer Congress, Topical Sessions, Toulouse, France, August 22-27, 2004*, pages 409–414, 2004. [Cited on page 94.]
- Marie-Luce Bourguet. Automatic Generation of Multimodal Interaction Models from Behavioural Data. HCI Workshop, 2006. [Cited on page 95.]
- Ronald J. Brachman and Hector J. Levesque. *Knowledge Representation and Reasoning*. The Morgan Kaufmann Series in Artificial Intelligence. Elsevier / Morgan Kaufmann, June 2004. [Cited on pages 119 and 121.]
- Ronald J. Brachman, Richard E. Fikes, and Hector J. Levesque. Krypton: A Functional Approach to Knowledge Representation. *Journal of Computer, IEEE Computer Society*, 16(10):67–73, October 1983. [Cited on page 119.]
- Michael E. Bratman. *Intentions, Plans, and Practical Reason*. Cambridge, MA: Harvard., 1987. [Cited on page 83.]
- Cynthia Breazeal, Andrew Brooks, Jesse Gray, Guy Hoffman, Cory Kidd, Hans Lee, Jeff Lieberman, Andrea Lockerd, and David Mulanda. Humanoid Robots as Cooperative Partners for People. *International Journal of Humanoid Robots*, 1(2):1–34, May 2004. [Cited on page 74.]
- Cynthia Breazeal. Social Robots for Health Applications. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2011, Boston, MA, USA, August 30 - September 3, 2011*, pages 5368–5371, 2011. [Cited on page 6.]
- Bert Bredeweg, Floris Linnebank, Anders Bouwer, and Jochem Liem. Garp3 - Workbench for Qualitative Modelling and Simulation. *Ecological Informatics. Journal of Ecological Informatics*, 4(5):263–281, November 2009. [Cited on page 230.]
- Bert Bredeweg, Jochem Liem, Wouter Beek, Floris Linnebank, Jorge Gracia, Esther Lozano, René Bühling Michael Wißner, Paulo Salles, Richard Noble, Andreas Zitek, Petya Borisova, and David Mioduser. DynaLearn - An Intelligent Learning Environment for Learning Conceptual Knowledge. *AI Magazine, Association for the Advancement of Artificial Intelligence*, 2013, 34(4):46–65, Winter Issue 2013. [Cited on page 230.]
- Susan E. Brennan and Maurice Williams. The Feeling of Another's Knowing: Prosody and Filled Pauses as Cues to Listeners about the Metacognitive States of Speakers. *Journal of Memory and Language*, 34(3):383–398, June 1995. [Cited on page 25.]
- Susan E. Brennan, Xin Chen, Christopher A. Dickinson, Mark B. Neider, and Gregory J. Zelinsky. Coordinating Cognition: The Costs and Benefits of Shared Gaze During Collaborative Search. *Journal of Cognition*, 106(3):1465–1477, March 2008. [Cited on pages 27, 29, 33, and 54.]
- Susan E. Brennan. The Grounding Problem in Conversations With and Through Computers. In S. R. Fussell and R. J. Kreuz, editors, *Social and Cognitive Psychological Approaches to Interpersonal Communication*, pages 201–225. Lawrence Erlbaum, Hillsdale, NJ, USA, 1998. [Cited on pages 15, 16, 22, 25, 59, 68, and 69.]
- Joost Broekens, Marcel Heerink, and Henk Rosendal. Assistive Social Robots in Elderly Care: A Review. *Journal of Gerontechnology*, 8(2):94–103, 2009. [Cited on page 6.]
- Douglas M. Brooks. The Teacher's Communicative Competence: The First Day of School. *Theory Into Practice, Special Issue on Classroom Communication/Verbal and Nonverbal*, 24(1):63–70, January 1985. [Cited on page 35.]
- Penelope Brown and Stephen C. Levinson. *Politeness: Some Universals in Language Usage*. Cambridge University Press, 1987. [Cited on pages 187 and 195.]
- David C. Brown, Stan C. Kwasny, B. Chandrasekaran, and Norman K. Sondheimer. An Experimental Graphics System with Natural Language Input. *Journal of Computers and Graphics*, 4(1):13–22, 1979. [Cited on page 89.]
- Ralf Bruns and Jürgen Dunkel. *Complex Event Processing - Komplexe Analyse von massiven Datenströmen mit CEP. Eine kompakte Einführung in die Grundprinzipien von Complex Event Processing (CEP)*. Springer Verlag, 2015. [Cited on page 118.]

- Jenny Brusik, Torbjörn Lager, Anna Hjalmarsson, and Preben Wik. DEAL: Dialogue Management in SCXML for Believable Game Characters. In Bill Kapralos, Michael Katchabaw, and Jay Rajnovich, editors, *Proceedings of the International Conference on Future Play, Future Play 2007, Toronto, Ontario, Canada, November 15 - 17, 2007*, pages 137–144. ACM, New York, NY, USA, 2007. [Cited on pages 86 and 133.]
- Paweł Budzianowski, Stefan Ultes, Pei-Hao Su, Nikola Mrkšić, Tsung-Hsien Wen, Iñigo Casanueva, Lina Rojas-Barahona, and Milica Gašić. Sub-domain Modelling for Dialogue Management with Hierarchical Reinforcement Learning. In *Proceedings of the 18th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), Saarbrücken, Germany, August 15-17, 2017*, pages 86–92. Association for Computational Linguistics, 2017. [Cited on page 84.]
- René Bühling, Michael Wißner, Markus Häring, Gregor U. Mehlmann, and Elisabeth André. Design Decisions for Virtual Characters in the DynaLearn Interactive Learning Environment. In *The Book of Abstracts of the 7th International Conference on Ecological Informatics, ISEI 2010, Ghent, Belgium, December 13-16, 2010*, pages 144–145, 2010. [Cited on page 231.]
- Harry Bunt, Jan Alexandersson, Jean Carletta, Jae-Woong Choe, Alex Chengyu Fang, Koiti Hasida, Kiyong Lee, Volha Petukhova, Andrei Popescu-Belis, Laurent Romary, Claudia Soria, and David Traum. Towards an ISO Standard for Dialogue Act Annotation. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the 7th International Conference on Language Resources and Evaluation, LREC 2010, Valletta, Malta, May 19-21*. European Language Resources Association (ELRA), 2010. [Cited on pages 62, 83, 99, and 161.]
- Harry Bunt, Michael Kipp, and Volha Petukhova. Using DiAML and ANVIL for Multimodal Dialogue Annotations. In *Proceedings of the 8th International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey*, pages 1301–1308. European Language Resources Association, 2012. [Cited on page 161.]
- Harry Bunt. The Semantics of Dialogue Acts. In *Proceedings of the 9th International Conference on Computational Semantics, IWCS 2011, Oxford, United Kingdom, January 12 - 14, 2011*, pages 1–13. Association for Computational Linguistics Stroudsburg, PA, USA, 2011. [Cited on pages 62, 83, 99, and 161.]
- George Butterworth. The Ontogeny and Phylogeny of Joint Visual Attention. In Andrew Whiten, editor, *Natural Theories of Mind: Evolution, Development, and Simulation of Everyday Mindreading*, pages 223–232. Basil Blackwell, Cambridge, MA, USA, 1991. [Cited on page 31.]
- Whitney L. Cade, Jessica L. Copeland, Natalie K. Person, and Sidney K. D’Mello. Dialogue Modes in Expert Tutoring. In *Proceedings of the 9th International Conference on Intelligent Tutoring Systems, ITS 2008, Montreal, Canada, June 23-27, 2008*, volume 5091 of *Lecture Notes in Computer Science*, pages 470–479, 2008. [Cited on page 231.]
- Angelo Cafaro, Nadine Glas, and Catherine Pelachaud. The Effects of Interrupting Behavior on Interpersonal Attitude and Engagement in Dyadic Interactions. In Catholijn M. Jonker, Stacy Marsella, John Thangarajah, and Karl Tuyls., editors, *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2016, Singapore, May 9–13, 2016*, pages 911–920. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, 2016. [Cited on pages 79 and 173.]
- Joseph N. Cappella. Mutual Influence in Expressive Behavior: Adult-Adult and Infant-Adult Dyadic Interaction. *Psychological Bulletin*, 89(1):101–132, January 1981. [Cited on pages 21 and 22.]
- Joseph N. Cappella. Behavioral and Judged Coordination in Adult Informal Social Interactions: Vocal and Kinesic Indicators. *Journal of Personality and Social Psychology*, 72(1):119–131, January 1997. [Cited on pages 19 and 21.]
- Jaime R. Carbonell. Mixed-Initiative Man-Computer Dialogues. BBN Report No 1971, Bolt, Beranek and Newman, Cambridge, MA, USA, 1970. [Cited on page 89.]
- Helen Carl. Nonverbal Communication during the Employment Interview. *ABCA Bulletin*, 44(4):14–19, December 1980. [Cited on page 233.]
- Lauri Carlson. *Dialogue Games - An Approach to Discourse Analysis*. Studies in Linguistics and Philosophy. D. Reidl Publishing Company, 1985. [Cited on page 83.]
- Berardina De Carolis, Catherine Pelachaud, Isabella Poggi, and Mark Steedman. APML, a Markup Language for Believable Behavior Generation. In *Life-Like Characters*, pages 65–85. Springer Berlin Heidelberg, 2004. [Cited on pages 96 and 115.]

- Robert L. Carpenter, Carl Pollard, and Alex Franz. The Specification and Implementation of Constraint-based Unification Grammars. In *Proceedings of the 2nd International Workshop on Parsing Technology, Cancun, Mexico, February 13-25, 1991*, pages 143–153. Sponsored by the Special Interest Group on Parsing of the Association for Computational Linguistics, 1991. [Cited on page 119.]
- Robert L. Carpenter. The Logic of Typed Feature Structures with Applications to Unification Grammars, Logic Programs and Constraint Resolution. In Cornelis Joost van Rijsbergen, editor, *Cambridge Tracts in Theoretical Computer Science 32*. Cambridge University Press, Cambridge, UK, 1992. [Cited on pages 11, 52, 63, 91, and 118.]
- Robert L. Carpenter. Skeptical and Credulous Unification with Applications to Lexical Templates and Inheritance. In Ted Briscoe, Ann Copestake, and Valeria de Paiva, editors, *Default Reasoning and Lexical Organization*. Cambridge University Press, Cambridge, UK, 1993. [Cited on page 119.]
- Maria Chiara Caschera, Arianna D’Ulizia, Fernando Ferri, and Patrizia Grifoni. Multimodal Systems: An Excursus of the Main Research Questions. In Ioana Ciuciu, Hervé Panetto, Christophe Debruyne, Alexis Aubry, Peter Bollen, Rafael Valencia-García, Alok Mishra, Anna Fensel, and Fernando Ferri, editors, *Proceedings of the Confederated International Workshops: OTM Academy, OTM Industry Case Studies Program, EI2N, FBM, INBAST, ISDE, META4eS, and MSC 2015, Rhodes, Greece, October 26-30, 2015*, pages 546–558. Springer International Publishing, 2015. [Cited on page 89.]
- Justine Cassell, Catherine Pelachaud, Norman Badler, Mark Steedman, Brett Achorn, Tripp Becket, Brett Derville, Scott Prevost, and Matthew Stone. Animated Conversation: Rule-based Generation of Facial Expression, Gesture and Spoken Intonation for Multiple Conversational Agents. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1994, Orlando, FL, USA, July 24 - 29, 1994*, pages 413–420. ACM, New York, NY, USA, 1994. [Cited on page 86.]
- Justine Cassell, Obed. E. Torres, and Scott. Prevost. Turn Taking versus Discourse Structure. In *Machine Conversations*, pages 143–153. Springer Science and Business Media, New York, 1999. [Cited on page 77.]
- Justine Cassell, Tim Bickmore, Lee Campbell, Hannes Vilhjálmsson, , and Hao Yan. Conversation as a System Framework: Designing Embodied Conversational Agents . In Justine Cassell, Joseph Sullivan, Scott Prevost, and Elizabeth F. Churchill, editors, *Embodied Conversational Agents*. MIT Press Cambridge, MA, USA, 2000. [Cited on page 82.]
- Justine Cassell, Joseph Sullivan, Scott Prevost, and Elizabeth F. Churchill. *Embodied Conversational Agents*. The MIT Press, Cambridge Massachusetts, London England, 2000. [Cited on pages 6, 15, 71, and 81.]
- Justine Cassell, Hannes Högni Vilhjálmsson, and Timothy Bickmore. BEAT: the Behavior Expression Animation Toolkit. In *Life-Like Characters, reprint from the Proceedings of SIGGRAPH’01, August 12-17, Los Angeles, CA*, pages 163–185. Springer Berlin Heidelberg, 2004. [Cited on pages 96, 99, and 115.]
- Marc Cavazza, Fred Charles, and Steven J. Mead. Agents’ Interaction in Virtual Storytelling. In Angélica de Antonio, Ruth Aylett, and Daniel Ballin, editors, *Proceedings of the 3rd International Conference on Intelligent Virtual Agents, IVA 2001, Madrid, Spain, September 10-11, 2001*, volume 2190 of *Lecture Notes in Computer Science*, pages 156–170, 2001. [Cited on pages 81 and 231.]
- Marc Cavazza, Fred Charles, and Steven J. Mead. Interacting with Virtual Characters in Interactive Storytelling. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2002, Bologna, Italy, July 15 - 19, 2002*, pages 318–325. ACM, New York, NY, USA, 2002. [Cited on page 84.]
- Marc Cavazza, Stéphane Donikian, Marc Christie, Ulrike Spierling, Nicolas Szilas, Peter Vorderer, Tilo Hartmann, Christoph Klimmt, Elisabeth André, Ronan Champagnat, Paolo Petta, and Patrick Olivier. The IRIS Network of Excellence: Integrating Research in Interactive Storytelling. In Ulrike Spierling and Nicolas Szilas, editors, *Proceedings of the 1st Joint International Conference on Interactive Digital Storytelling, ICIDS 2008, Erfurt, Germany, November 26-29, 2008*, pages 14–19, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. [Cited on pages 81, 231, and 232.]
- J. K. Chambers, Peter Trudgill, and Natalie Schilling-Estes. *The Handbook of Language Variation and Change*. Blackwell Publishing Handbooks in Linguistics, 2004. [Cited on page 71.]
- Jaeseung Chang and Marie-Luce Bourguet. Usability Framework for the Design and Evaluation of Multimodal Interaction. In David England, editor, *Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction - Volume 2, BCS-HCI 2008, Liverpool, United Kingdom, 1-5 September 2008*, pages 123–126. BCS Learning and Development Ltd., Swindon, UK, 2008. [Cited on page 95.]

- Crystal Chao and Andrea Thomaz. Timing in Multimodal Turn-Taking Interactions: Control and Analysis using Timed Petri Nets. *Journal of Human-Robot Interaction*, 1(1):4–25, 2011. [Cited on pages 79, 87, 99, and 133.]
- Crystal Chao and Andrea Thomaz. Controlling Social Dynamics with a Parametrized Model of Floor Regulation. *Journal of Human-Robot Interaction*, 2(1):4–29, 2013. [Cited on pages 80 and 87.]
- Crystal Chao and Andrea Thomaz. Timed Petri Nets for Fluent Turn-Taking over Multimodal Interaction Resources in Human-Robot Collaboration. *International Journal of Robotics Research*, 35(11):1330–1353, September 2016. [Cited on pages 87, 99, and 133.]
- Crystal Chao, Justin Smith, and Andrea L. Thomaz. CADENCE for Collaboration and Companionship with Robots. In *2014 AAAI Fall Symposium Series*, 2014. [Cited on pages 80 and 87.]
- Crystal Chao. Timing Multimodal Turn-Taking for Human-Robot Cooperation. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction, ICMI 2012, Santa Monica, California, USA, October 22-26, 2012*, ACM International Conference Proceedings, pages 309–312. ACM, New York, NY, USA, 2012. [Cited on pages 80 and 87.]
- Crystal Chao. *Timing Multi-Modal Turn-Taking in Human-Robot Cooperative Activity*. PhD thesis, School of Interactive Computing, Georgia Institute of Technology, 2015. [Cited on page 87.]
- Tanya L. Chartrand and John A. Bargh. The Chameleon Effect: The Perception-Behavior Link and Social Interaction. *Journal of Personality and Social Psychology*, 76(6):893–910, June 1999. [Cited on pages 18 and 37.]
- Tanya L. Chartrand and Jessica L. Lakin. The Antecedents and Consequences of Human Behavioral Mimicry. *Annual Review of Psychology*, 64:285–308, 2013. [Cited on pages 17, 18, 21, 28, 29, 37, and 182.]
- Tanya L. Chartrand and Rick van Baaren. Human Mimicry. In *Advances in Experimental Social Psychology*, volume 41, chapter 5, pages 219–274. Elsevier Inc, 2009. [Cited on page 18.]
- Sonu Chopra-Khullar and Norman I. Badler. Where To Look? Automating Attending Behaviors of Virtual Human Characters. *Autonomous Agents and Multi-Agent Systems*, 4(1-2):16–23, May 1999. [Cited on page 37.]
- Patricia Clancy. Analysis of a Conversation. *Anthropological Linguistics*, 14(3):78–86, March 1972. [Cited on page 40.]
- Herbert H. Clark and Susan E. Brennan. Grounding in Communication. In Lauren B. Resnick, John M. Levine, and Stephanie D. Teasley, editors, *Perspectives on Socially Shared Cognition*, pages 222–233. American Psychological Association, Washington, DC, USA, 1991. [Cited on pages 15, 16, 22, 25, 26, 44, 54, 59, 60, 68, 69, and 71.]
- Herbert H. Clark and Thomas B. Carlson. Hearers and Speech Acts. *Journal of Language*, 58(2):332–373, June 1982. [Cited on pages 33 and 183.]
- Herbert H. Clark and Meredyth A. Krych. Speaking while Monitoring Addressees for Understanding. *Journal of Memory and Language*, 50(1):62–81, January 2004. [Cited on pages 23, 24, 26, 27, 54, 60, and 62.]
- Herbert H. Clark and Catherine K. Marshall. Definite Reference and Mutual Knowledge. In A. K. Joshi, B. L. Webber, and I. A. Sag, editors, *Elements of Discourse Understanding*. Cambridge University Press, 1981. [Cited on pages 31 and 69.]
- Herbert H. Clark and Edward F. Schaefer. Collaborating on Contributions to Conversations. *Language and Cognitive Processes*, 2(1):19–41, October 1987. [Cited on page 23.]
- Herbert H. Clark and Edward F. Schaefer. Contributing to Discourse. *Journal of Cognitive Science*, 13(2):259–294, April-June 1989. [Cited on pages 16, 23, and 26.]
- Herbert H. Clark and Edward F. Schaefer. Dealing with Overhearers. In H. H. Clark, editor, *Arenas of Language Use*, pages 248–274. University of Chicago Press, Chicago, USA, 1992. [Cited on page 32.]
- Herbert H. Clark and Deanna Wilkes-Gibbs. Referring as a Collaborative Process. *Journal of Cognitive Science*, 22(1):1–39, February 1986. [Cited on pages 5, 16, 23, 26, 34, and 71.]
- Herbert H. Clark. *Using Language*. Cambridge University Press, Cambridge, UK, May 1996. [Cited on pages 5, 16, 17, 22, 23, 24, 26, 32, 33, 34, 35, 42, 59, 70, 71, 183, and 235.]
- Herbert H. Clark. Coordinating with Each Other in a Material World. *Discourse Studies*, SAGE Publications, London, Thousand Oaks, CA and New Delhi, 7(4-5):507–525, October 2005. [Cited on pages 5, 15, 16, 22, 27, 32, 33, 34, 42, 54, 60, 62, 69, and 71.]

- William F. Clocksin and Christopher S. Mellish. *Programming in Prolog*. Springer-Verlag, Berlin-Heidelberg-New York, 1981. [Cited on pages 11, 52, 67, 81, 88, and 93.]
- Jennifer Coates. No Gaps, Lots of Overlaps: Turn-Taking patterns in the Talk of Women Friends'. In David Graddol, Janet Maybin, and Barry Stierer, editors, *Researching Language and Literacy in Social Context*, pages 177–192. Clevedon, UK: Multilingual Matters, 1994. [Cited on page 42.]
- Philip R. Cohen, Mary Dalrymple, Douglas B. Moran, Fernando C.N. Pereira, Joseph W. Sullivan, Robert A. Gargan Jr., Jon L. Schlossberg, and Sherman W. Tyler. Synergistic Use of Direct Manipulation and Natural Language. In K. Bice and C. Lewis, editors, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 1989, Austin, Texas, USA, April 30 - June 4, 1989*, pages 227–233. ACM, New York, NY, USA, 1989. [Cited on page 89.]
- Philip R. Cohen, Michael Johnston, David McGee, Sharon Oviatt, Jay Pittman, Ira Smith, Liang Chen, and Josh Clow. QuickSet: Multimodal Interaction for Simulation Set-up and Control. In *Proceedings of the 5th Conference on Applied Natural Language Processing, Washington, DC, USA, March 31- April 3, 1997*, pages 20–24. Association for Computational Linguistics, 1997. [Cited on pages 91 and 118.]
- Philip R. Cohen, Michael Johnston, David McGee, Sharon Oviatt, Jay Pittman, Ira Smith, Liang Chen, and Josh Glow. QuickSet: Multimodal Interaction for Distributed Applications. In *Proceedings of the 5th ACM International Conference on Multimedia, MULTIMEDIA 1997, Seattle, Washington, USA, November 9-13, 1997*, pages 31–40, 1997. [Cited on pages 91 and 119.]
- Philip R. Cohen, Michael Johnston, David McGee, Sharon L. Oviatt, Josh Clow, and Ira A. Smith. The Efficiency of Multimodal Interaction: A Case Study. In *Proceedings of the 5th International Conference on Spoken Language Processing, Incorporating The 7th Australian International Speech Science and Technology Conference, Sydney, Australia, November 30 - December 4, 1998*, pages 1–4, 1998. [Cited on page 92.]
- Philip Cohen, David McGee, and Josh Clow. The Efficiency of Multimodal Interaction for a Map-based Task. In *Proceedings of the 6th Conference on Applied Natural Language Processing, ANCL 2000*, pages 331–338. Association for Computational Linguistics, Stroudsburg, PA, USA, 2000. [Cited on page 92.]
- Philip R. Cohen. Integrated Interfaces for Decision Support with Simulation. In Barry L. Nelson, W. David Kelton, and Gordon M. Clark, editors, *Proceedings of the 1991 Winter Simulation Conference, WCS 1991*, pages 1066–1072. IEEE Computer Society Washington, DC, USA, 1991. [Cited on page 89.]
- Alain Colmerauer. Metamorphosis Grammars. In Leonard Bolc, editor, *Natural Language Communication with Computers*, volume Lecture Notes in Computer Science of 63, pages 133–189. Springer Berlin Heidelberg, 1978. [Cited on pages 67 and 130.]
- William S. Condon and William D. Ogston. Sound Film Analysis of Normal and Pathological Behavior Patterns. *Journal of Nervous and Mental Disease*, 143(4):338–347, October 1966. [Cited on pages 17 and 18.]
- William S. Condon and William D. Ogston. A Segmentation of Behavior. *Journal of Psychiatric Research*, 5(3):221–235, September 1967. [Cited on pages 17 and 18.]
- Mark Cook and Jacqueline M. C. Smith. The Role of Gaze in Impression Formation. *British Journal of Social and Clinical Psychology*, 14(1):19–25, February 1975. [Cited on pages 30 and 38.]
- Robin Cooper, Ian Lewin, and Alan W. Black. *Prolog and Natural Language Semantics*. Lecture Notes for Computational Semantics. Department of Artificial Intelligence, University of Edinburgh, 1993. [Cited on page 130.]
- Mark Core and James Allen. Coding Dialogs with the DAMSL Annotation Scheme. In *Working Notes of AAAI Fall Symposium on Communicative Action in Humans and Machines, Cambridge, MA, USA, November 8-10, 1997*. AAAI Press, 1997. [Cited on pages 99 and 232.]
- Richard G. Coss. The Perceptual Aspects of Eye-Spot Patterns and their Relevance to Gaze Behaviour. In C. Hutt and S. J. Hutt, editors, *Behaviour Studies in Psychiatry*, pages 12–147. Pergamon Press, London, UK, 1970. [Cited on page 31.]
- Joëlle Coutaz, Laurence Nigay, Daniel Salber, Ann Blandford, Jon May, and Richard M. Young. Four Easy Pieces for Assessing the Usability of Multimodal Interaction: The CARE Properties. In K. Nordby, P. H. Helmersen, D. J. Gilmore, and S. A. Arnesen, editors, *Proceedings of the IFIP TC13 Fifth International Conference on Human-Computer Interaction, INTERACT 1995, Lillehammer, Norway, June 25-29, 1995*, pages 115–120. Chapman and Hall, 1995. [Cited on page 90.]

- Michael A. Covington. GULP 3.1: An Extension of Prolog for Unification-Based Grammar. *Artificial intelligence Research Report AI-1994-06*, Artificial Intelligence Center, The University of Georgia, May 1994. [Cited on page 120.]
- Michelle L. Crane and Juergen Dingel. UML vs. Classical vs. Rhapsody Statecharts: Not All Models are Created Equal. *Journal of Software and Systems Modeling*, 6(4):415–435, December 2007. [Cited on pages 86 and 205.]
- Nigel Crook, Cameron Smith, Marc Cavazza, Stephen Pulman, Roger Moore, and Johan Boye. Handling user Interruptions in an Embodied Conversational Agent. In *Proceedings of the 9th International Conference on Autonomous Agents and Multi-agent Systems, AAMAS 2010, International Workshop on Interacting with ECAs as Virtual Characters, Toronto, Canada, May, 10-14, 2010*, pages 27–33, 2010. [Cited on pages 79 and 80.]
- Nigel Crook, Debora Field, Cameron Smith, Sue Harding, Stephen Pulman, Marc Cavazza, Daniel Charlton, Roger Moore, and Johan Boye. Generating Context-Sensitive ECA Responses to User Barge-In Interruptions. *Journal on Multimodal User Interfaces*, 6(13):13–25, April 2012. [Cited on page 99.]
- Cynthia L. Crown and Debora A. Cummins. Objective Versus Perceived Vocal Interruptions in the Dialogues of Unacquainted Pairs, Friends, and Couples. *Journal of Language and Social Psychology*, 17(3):372–389, September 1998. [Cited on page 173.]
- Cynthia L. Crown, Stanley Feldstein, Michael D. Jasnow, Beatrice Beebe, and Joseph Jaffe. The Cross-Modal Coordination of Interpersonal Timing: Six-Week-Olds Infants' Gaze with Adults' Vocal Behavior. *Journal of Psycholinguistic Research*, 31(1):1–23, January 2002. [Cited on pages 5 and 21.]
- Cynthia L. Crown. Coordinated Interpersonal Timing of Vision and Voice as a Function of Interpersonal Attraction. *Journal of Language and Social Psychology*, 10(1):29–46, March 1991. [Cited on pages 21 and 22.]
- Fred Cummins. Gaze and Blinking in Dyadic Conversation: A Study in Coordinated Behaviour Among Individuals. *Journal of Language and Cognitive Processes*, 27(10):1525–1549, December 2012. [Cited on page 35.]
- Ken Currie and Austin Tate. O-Plan: The Open Planning Architecture. *Journal of Artificial Intelligence*, 52(1):49–86, November 1991. [Cited on page 83.]
- Rogério E. da Silva, Ido A. Iurgel, and Manuel F. dos Santos. Towards Virtual Actors - The Next Step for the Entertainment Industry. In *Proceedings of the 8th Brazilian Symposium on Games and Digital Entertainment, Rio de Janeiro, RJ, Brazil, October 8-10, 2009*, pages 105–108, 2009. [Cited on page 82.]
- James M. Dabbs. Similarity of Gestures and Interpersonal Influence. *Proceedings of the 77th Annual Convention of the American Psychological Association, Washington D.C., USA*, 4(1):337–339, 1969. [Cited on page 19.]
- Martyn Dade-Robertson. Visual Scenario Representation in the Context of a Tool for Interactive Storytelling. In M. Cavazza and S. Donikian, editors, *Proceedings of the 4th International Conference on Virtual Storytelling: Using Virtual Reality Technologies for Storytelling, ICVS 2007, Saint-Malo, France, December 5 - 7, 2007*. Springer-Verlag Berlin, Heidelberg, 2007. [Cited on page 88.]
- Duilio D'Alfonso. Generalized Quantifiers: Logic and Language. *Logic and Philosophy of Science*, 9(1):85–94, 2011. [Cited on page 130.]
- Ionut Damian, Birgit Endrass, Peter Huber, Nikolaus Bee, and Elisabeth André. Individualized Agent Interactions. In Jan M. Allbeck and Petros Faloutsos, editors, *Proceedings of the 4th International Conference on Motion in Games, MIG 2011, Edinburgh, UK, November 13-15, 2011*, volume 7060 of *Lecture Notes in Computer Science*, pages 15–26. Springer Berlin Heidelberg, 2011. [Cited on page 232.]
- Ionut Damian, Tobias Baur, Patrick Gebhard, Kaska Porayska-Pomsta, and Elisabeth André. A Software Framework for Social Cue-based Interaction with a Virtual Recruiter. In Ruth Aylett, Brigitte Krenn, Catherine Pelachaud, and Hiroshi Shimodaira, editors, *Proceedings of the 13th International Conference on Intelligent Virtual Agents, IVA 2013, Edinburgh, UK, August 29-31, 2013*, volume 8108 of *Lecture Notes in Computer Science*, pages 444–445. Springer, Berlin, Heidelberg, 2013. [Cited on pages 151, 155, and 233.]
- Ionut Damian, Tobias Baur, Birgit Lugin, Patrick Gebhard, Gregor U. Mehlmann, and Elisabeth André. Games are Better than Books: In-Situ Comparison of an Interactive Job Interview Game with Conventional Training. In Cristina Conati, Neil Heffernan, Antonija Mitrovic, and M. Felisa Verdejo, editors, *Proceedings of the 17th International Conference on Artificial Intelligence in Education, AIED 2015, Madrid, Spain, June 22-26, 2015*, volume 9112 of *Lecture Notes in Computer Science*, pages 84–94. Springer International Publishing, Switzerland, 2015. [Cited on pages 6, 151, 155, 233, and 234.]

- Belur V. Dasarathy. Sensor Fusion Potential Exploitation - Innovative Architectures and Illustrative Applications. *Proceedings of the IEEE*, 85(1):24–38, 1997. [Cited on page 63.]
- Martha Davis. Introduction. In Martha Davis, editor, *Interaction Rhythms: Periodicity in Communicative Behavior*, pages 23–29. New York: Human Sciences Press, 1982. [Cited on pages 19 and 59.]
- Emilie Delaherche, Mohamed Chetouani, Ammar Mahdhaoui, Catherine Saint-Georges, Sylvie Viaux, and David Cohen. Interpersonal Synchrony: A Survey of Evaluation Methods across Disciplines. *IEEE Transactions on Affective Computing*, 3(3):349–365, 2012. [Cited on pages 29 and 59.]
- Wendy Despain. *Writing for Video Games: From FPS to RPG*. A K Peters/CRC Press, 2008. [Cited on page 84.]
- Kathryn Dindia. The Effects of Sex of Subject and Sex of Partner on Interruptions. *Journal of Human Communication Research*, 13(3):345–371, March 1987. [Cited on pages 40 and 44.]
- Allen T. Dittmann and Lynn G. Llewellyn. The Phonemic Clause as a Unit of Speech Decoding. *Journal of Personality and Social Psychology*, 6(3):341–349, July 1967. [Cited on pages 21 and 43.]
- Sidney D’Mello and Art Graesser. AutoTutor and Affective Autotutor: Learning by Talking with Cognitively and Emotionally Intelligent Computers That Talk Back. *ACM Transactions on Interactive Intelligent Systems*, 2(4):1–39, January 2013. [Cited on pages 85 and 229.]
- Gwyneth Doherty-Sneddon and Fiona G. Phelps. Gaze Aversion: A Response to Cognitive or Social Difficulty? *Memory and Cognition*, Springer-Verlag, 33(4):727–733, June 2005. [Cited on pages 5, 37, and 76.]
- Marek W. Doniec, Ganghua Sun, and Brian Scassellati. Active Learning of Joint Attention. In *Proceedings of the 6th IEEE-RAS International Conference on Humanoid Robots, Genova, Italy, December 4-6, 2006*, pages 34–39, 2006. [Cited on page 74.]
- Kent Drummond. A Backward Glance at Interruptions. *Western Journal of Communication: Sequential Organization of Conversational Activities*, 53(2):150–166, 1989. [Cited on pages 15, 39, 40, and 173.]
- Doron Drusinsky. Visual Formal Specification using (N)TLCharts: Statechart Automata with Temporal Logic and Natural Language Conditioned Transitions. In *Proceedings of the 18th International Parallel and Distributed Processing Symposium, IPDPS 2004, Santa Fe, New Mexico, April 26-30, 2004*, pages 268–276, 2004. [Cited on pages 86, 201, and 205.]
- Doron Drusinsky. *Modeling and Verification Using UML Statecharts: A Working Guide to Reactive System Design, Runtime Monitoring and Execution-based Model Checking*. Newnes Publishers, 2006. [Cited on pages 86, 201, and 205.]
- Bruno Dumas, Denis Lalanne, and Rolf Ingold. HephaisTK: A Toolkit for Rapid Prototyping of Multimodal Interfaces. In James L. Crowley, Yuri A. Ivanov, Christopher Richard Wren, Daniel Gatica-Perez, Michael Johnston, and Rainer Stiefelhagen, editors, *Proceedings of the 11th International Conference on Multimodal Interfaces and the Sixth Workshop on Machine Learning for Multimodal Interaction, ICMI-MLMI 2009, Cambridge, Massachusetts, USA, November 2-6, 2009*, pages 231–232. ACM, New York, NY, USA, 2009. [Cited on pages 64, 82, 93, and 99.]
- Bruno Dumas, Denis Lalanne, and Sharon Oviatt. Multimodal Interfaces: A Survey of Principles, Models and Frameworks. In Denis Lalanne and Jürg Kohlas, editors, *Human Machine Interaction*, volume 5440 of *Lecture Notes in Computer Science*, pages 3–26. Springer Berlin Heidelberg, 2009. [Cited on pages 62, 81, 82, and 93.]
- Bruno Dumas, Denis Lalanne, and Rolf Ingold. Description Languages for Multimodal Interaction: A Set of Guidelines and its Illustration with SMUIML. *Journal on Multimodal User Interfaces*, 3(3):237–247, April 2010. [Cited on pages 81, 82, and 93.]
- Bruno Dumas, Beat Signer, and David Lalanne. A Graphical Editor for the SMUIML Multimodal User Interaction Description Language. *Science of Computer Programming*, 86:30–42, June 2014. [Cited on pages 82, 93, and 118.]
- Starkey Duncan and Donald W. Fiske. *Face-To-Face Interaction: Research, Methods, and Theory*. Routledge Library Editions: Communication Studies, 1977. [Cited on pages 5, 35, and 38.]
- Starkey Duncan and George Niederehe. On Signalling That It’s Your Turn to Speak. *Journal of Experimental Social Psychology*, 10(3):234–247, May 1974. [Cited on page 36.]

- Starkey Duncan. Some Signals and Rules for Taking Speaking Turns in Conversations. *Journal of Personality and Social Psychology*, 23(2):283–292, August 1972. [Cited on pages 5, 17, 21, 35, 42, 43, 55, 77, 80, and 235.]
- Starkey Duncan. Some Signals and Rules for Taking Speaking Turns in Conversations. In Shirley Weitz, editor, *Nonverbal Communication: Readings with Commentary*, pages 298–311. Oxford University Press, 1974. [Cited on pages 21, 35, 55, 77, and 171.]
- Jay Earley. An Efficient Context-Free Parsing Algorithm. *Communications of the ACM*, 13(2):94–102, February 1970. [Cited on page 92.]
- Carole Edelsky. Who's Got the Floor? *Journal of Language in Society*, 10(3):383–421, 1981. [Cited on pages 42 and 43.]
- Patrick Ehlen and Michael Johnston. A Multimodal Dialogue Interface for Mobile Local Search. In *Proceedings of the Companion Publication of the 2013 International Conference on Intelligent User Interfaces Companion, IUI 2013 Companion, Santa Monica, California, USA, March 19–22, 2013*, pages 63–64. ACM, New York, NY, USA, 2013. [Cited on pages 91 and 119.]
- Howard Ehrlichman and Dragana Micic. Why Do People Move Their Eyes When They Think? *Current Directions in Psychological Science*, 21(2):96–100, April 2012. [Cited on page 37.]
- Paul Ekman and Wallace V. Friesen. Nonverbal Leakage and Clues to Deception. *Journal of Psychiatry*, 32(1):88–106, February 1969. [Cited on page 25.]
- Paul Ekman and Wallace V. Friesen. *Unmasking the Face: A Guide to Recognizing Emotions from Facial Clues*. Malor Books, Los Altos, California, USA, 2003. [Cited on page 19.]
- Paul Ekman. Nebraska Symposium on Motivation. In J. Cole, editor, *Universal and Cultural Differences in Facial Expressions of Emotion*, pages 207–283. University of Nebraska-Lincoln Press, Lincoln, Nebraska, USA, 1972. [Cited on page 19.]
- Paul Ekman. Nonverbal Behavior and Communication. In Stanley Feldstein and Aron W. Siegman, editors, *Facial Expressions*, pages 97–116. Lawrence Erlbaum Association, New Jersey, USA, 1977. [Cited on pages 19 and 25.]
- Maarten H. Van Emden and Robert A. Kowalski. The Semantics of Predicate Logic As a Programming Language. *Journal of the ACM*, 23(4):733–742, October 1976. [Cited on pages 65, 67, and 121.]
- Nathan J. Emery. The Eyes Have It: The Neuroethology, Function and Evolution of Social Gaze. *Neuroscience and Biobehavioral Reviews*, 24(6):581–604, August 2000. [Cited on pages 31, 32, and 34.]
- Birgit Endrass, Michael Wißner, Gregor Mehlmann, René Bühling, Markus Häring, and Elisabeth André. Teenage Girls as Authors for Digital Storytelling - A Practical Experience Report. In *Workshop on Education in Interactive Digital Storytelling held on the 3rd International Conference on Interactive Digital Storytelling, ICIDS 2010, Edinburgh, UK, November 1 - 3, 2010*, 2010. [Cited on page 239.]
- Birgit Endrass, Christoph Klimmt, Gregor U. Mehlmann, Elisabeth André, and Christian Roth. Exploration of User Reactions to Different Dialog-based Interaction Styles. In *Proceedings of the 4th International Conference on Interactive Digital Storytelling, ICIDS 2011, Vancouver, Canada, November 28 - 1 December, 2011*, volume 7069 of *Lecture Notes in Computer Science*, pages 243–248. Springer-Verlag, Berlin, Heidelberg, 2011. [Cited on page 233.]
- Ralf Engel. Robust and Efficient Semantic Parsing of FreeWord Order Languages in Spoken Dialogue Systems. In *Proceedings of 9th Conference on Speech Communication and Technology and 6th Interspeech, ISCA 2005, Lisbon, Portugal, September 4–8, 2005*, pages 1–4. International Speech Communication Association, 2005. [Cited on page 232.]
- Kutluhan Erol, James Hendler, and Dana S. Nau. HTN Planning: Complexity and Expressivity. In *Proceedings of the 12th National Conference on Artificial Intelligence, AAAI 1994, Menlo Park, CA, USA*, pages 1123–1128. American Association for Artificial Intelligence, 1994. [Cited on page 83.]
- Anita Esposito. Sex Differences in Children's Conversation. *Journal of Language and Speech*, 22(3):213–220, July 1979. [Cited on page 39.]
- Ralph V. Exline and Clarence Eldridge. Effects of Two Patterns of a Speaker's Visual Behavior upon the Perception of the Authenticity of his Verbal Message. In *Annual Meeting of the Eastern Psychological Association Convention, Boston, MA, USA, April 6–8, 1967*. [Cited on page 35.]

- Ralph V. Exline and L. C. Winters. Affective Relations and Mutual Glances in Dyads. In S. S. Tomkins and C. Izard, editors, *Affect, Cognition and Personality*. Tavistock, London, UK, 1966. [Cited on page 38.]
- Ralph V. Exline, David Gray, and Dorothy Schuette. Visual Behavior in a Dyad as Affected by Interview Content and Sex of Respondent. *Journal of Personality and Social Psychology*, 1(13):201–209, March 1965. [Cited on pages 37 and 38.]
- Rui Fang, Joyce Y. Chai, and Fernanda Ferreira. Between Linguistic Attention and Gaze Fixations Inmultimodal Conversational Interfaces. In James L. Crowley, Yuri A. Ivanov, Christopher Richard Wren, Daniel Gatica-Perez, Michael Johnston, and Rainer Stiefelhagen, editors, *Proceedings of the 11th International Conference on Multimodal Interfaces and the Sixth Workshop on Machine Learning for Multimodal Interaction, ICMI-MLMI 2009, Cambridge, Massachusetts, USA, November 2-6, 2009*, pages 143–150. ACM, New York, NY, USA, 2009. [Cited on page 66.]
- Stanley Feldstein and Joan Welkowitz. Nonverbal Behavior and Communication. In Aron W. Siegman and Stanley Feldstein, editors, *A Chronography of Conversation: In Defense of an Objective Approach*, pages 435–499. Hillsdale, NJ, England: Lawrence Erlbaum Associates, Inc., 1987. [Cited on pages 22 and 40.]
- Stanley Feldstein, Joseph Jaffe, Beatrice Beebe, Cynthia L. Crown, Michael Jasnow, Harold Fox, and Sharon Gordon. Coordinated Interpersonal Timing in Adult-Infant Vocal Interactions: A Cross-Site Replication. *Infant Behavior and Development*, 16(4):455–470, October–December 1993. [Cited on pages 21 and 22.]
- Nicola Ferguson. Simultaneous Speech, Interruptions, and Dominance. *British Journal of Clinical Psychology*, 16(4):295–302, November 1977. [Cited on pages V, 40, 41, 60, 173, 175, 187, and 195.]
- Fernanda Ferreira. Syntax in Language Production: An Approach Using Tree-Adjoining Grammars. In L. Wheelodon, editor, *Aspects of Language Production*, pages 291–330. Psychology Press Taylor and Francis, Philadelphia, PA, USA, 2000. [Cited on page 23.]
- Richard E. Fikes and Tom Kehler. The Role of Frame-based Representation in Reasoning. *Communications of the ACM*, ACM, New York, NY, USA, 28(9):904–920, September 1985. [Cited on pages 90 and 119.]
- Richard E. Fikes and Nils J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Journal of Artificial Intelligence*, 2(3-4):189–208, Winter 1971. [Cited on page 82.]
- Kerstin Fischer, Lars C. Jensen, Franziska Kirstein, Sebastian Stabinger, Özgür Erkent, Dadhichi Shukla, and Justus Piater. The Effects of Social Gaze in Human-Robot Collaborative Assembly. In Adriana Tapus, Elisabeth Abdré, Jean-Claude Martin, François Ferland, and Mehdi Ammi, editors, *Proceedings of the 7th International Conference on Social Robotics, ICSR 2015, Paris, France, October 26-30, 2015, Springer International Publishing Switzerland*, volume 9388 of *Lecture Notes in Artificial Intelligence*, pages 204–213, 2015. [Cited on page 30.]
- Pamela M. Fishman. Interaction: The Work Women Do. In Nikolas Coupland and Adam Jaworski, editors, *Sociolinguistics: A Reader*, pages 416–429. London, UK: Macmillan Education, London, 1997. [Cited on pages 17 and 43.]
- Gerhard Fliedner and Daniel Bobbert. DiaMant: A Tool for Rapidly Developing Spoken Dialogue Systems. In *Proceedings of the 7th Workshop on the Semantics and Pragmatics of Dialogue, DiaBruck 2003, Wallerfangen, Germany, September 4-6, 2003*, 2003. [Cited on page 85.]
- Annika Flycht-Eriksson and Arne Jönsson. Dialogue and Domain Knowledge Management in Dialogue Systems. In *Proceedings of the 1st SIGdial Workshop on Discourse and Dialogue, Hong Kong, October 7-8, 2000*, pages 121–130, 2000. [Cited on page 69.]
- Annika Flycht-Eriksson. A Survey of Knowledge Sources in Dialogue Systems. In *Proceedings of Workshop on Knowledge and Reasoning in Practical Dialogue Systems, held at the 16th International Joint Conference on Artificial Intelligence, IJCAI 1999, Seattle, Washington, USA, August 5-10, 1999*, pages 41–48, 1999. [Cited on page 69.]
- Jerry Alan Fodor. *The Modularity of Mind. An Essay on Faculty Psychology*. The MIT Press, Cambridge, MA, USA, 1983. [Cited on page 141.]
- Terrence Fong, Illah Nourbakhsh, and Kerstin Dautenhahn. A Survey of Socially Interactive Robots. *Journal of Robotics and Autonomous Systems*, 42(3-4):143–166, March 2003. [Cited on pages 6, 15, 63, and 68.]

- Charles L. Forgy. Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem. In Peter G. Raeth, editor, *Expert Systems*, pages 324–341. IEEE Computer Society Press, Los Alamitos, CA, USA, 1990. [Cited on page 81.]
- Ira R. Forman and Nate Forman. *Java Reflection in Action*. Manning Publications, Greenwich, USA, 2004. [Cited on page 205.]
- Lyn Frazier and Charles Clifton. Construal: Overview, Motivation, and Some New Evidence. *Journal of Psycholinguistic Research*, 26(3):277–295, May 1997. [Cited on page 23.]
- Reva Freedman. Plan-based Dialogue Management in a Physics Tutor. In *Proceedings of the 6th Conference on Applied Natural Language Processing, ANLC 2000, Seattle, Washington, USA, April 29 - May 4, 2000*, pages 52–59. Association for Computational Linguistics, Stroudsburg, PA, USA, 2000. [Cited on page 83.]
- Russell M. Freeman, Simon J. Julier, and Anthony J. Steed. A Method for Predicting Marker Tracking Error. In *Proceedings of the 11th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR 2007, Nara, Japan, November 13-16, 2007*, 2007. [Cited on pages 66 and 159.]
- Ernest Friedman-Hill. *Jess in Action: Rule Based Systems in Java*. Manning Publications, 2003. [Cited on pages 81 and 93.]
- Alexandra Frischen, Andrew P. Bayliss, and Steven P. Tipper. Gaze Cueing of Attention: Visual Attention, Social Cognition, and Individual Differences. *Psychological Bulletin*, 133(4):694–724, July 2007. [Cited on page 33.]
- Rick Fry and Gene F. Smith. The Effects of Feedback and Eye Contact on Performance of a Digit-Encoding Task. *Journal of Social Psychology*, 96(1):145–146, February 1975. [Cited on page 35.]
- Atsushi Fukayama, Takehiko Ohno, Naoki Mukawa, Minako Sawaki, and Norihiro Hagita. Messages Embedded in Gaze of Interface Agents - Impression Management with Agent's Gaze. In *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2002, Minneapolis, Minnesota, USA, April 20 - 25, 2002*, pages 41–48. ACM, New York, NY, USA, 2002. [Cited on pages 78 and 193.]
- Malte Gabsdil. Clarification in spoken dialogue systems. In *Proceedings of the AAAI Spring Symposium on Natural Language Generation in Spoken and Written Dialogue, Stanford University, Palo Alto, California, USA, March 24-26, 2003*, pages 28–35. AAAI, American Association for Artificial Intelligence, 2003. [Cited on page 70.]
- Sudeep Gandhe and David R. Traum. An Evaluation Understudy for Dialogue Coherence Models. In *Proceedings of the 9th Workshop on Discourse and Dialogue, SIGDIAL 2008, Columbus, Ohio, USA, June 19-20, 2008*, pages 172–181. Association for Computational Linguistics, Morristown, NJ, USA, 2008. [Cited on pages 60 and 86.]
- Sudeep Gandhe, David DeVault, Antonio Roque, Bilyana Martinovski, Ron Artstein, Anton Leuski, Jillian Gerten, and David Traum. From Domain Specification to Virtual Humans: An Integrated Approach to Authoring Tactical Questioning Characters. In *Proceedings of the 9th Annual Conference of the International Speech Communication Association, INTERSPEECH 2009, Brisbane, Australia, September 22-26, 2008*, pages 1–4, 2008. [Cited on page 86.]
- Sudeep Gandhe, Nicolle Whitman, David Traum, and Ron Artstein. An Integrated Authoring Tool for Tactical Questioning Dialogue Systems. In *Proceedings of the 6th IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, KRPDS 2009, Pasadena, California, USA, July 11-17, 2009*, page 10, 2009. [Cited on pages 86 and 99.]
- Patrick Gebhard and Martin Klesen. Using Real Objects to Communicate with Virtual Characters. In Themis Panayiotopoulos, Jonathan Gratch, Ruth Aylett, Daniel Ballin, Patrick Olivier, and Thomas Rist, editors, *Proceedings of the 5th International Working Conference on Intelligent Virtual Agents, IVA 2005, Kos, Greece, September 12-14, 2005*, volume 3661 of *Lecture Notes in Computer Science*, pages 99–110. Springer-Verlag, Berlin, Heidelberg, 2005. [Cited on page 199.]
- Patrick Gebhard, Michael Kipp, Martin Klesen, and Thomas Rist. Authoring Scenes for Adaptive, Interactive Performances. In *Proceedings of the 21st International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2003, Melbourne, Victoria, Australia, July 14-18, 2003*, International Conference on Autonomous Agents, pages 725–732. ACM, New York, NY, USA, 2003. [Cited on pages 85, 108, 109, 118, 133, and 201.]
- Patrick Gebhard, Michael Kipp, Martin Klesen, and Thomas Rist. What Are They Going to Talk About? Towards Life-Like Characters that Reflect on Interactions with Users. In *Proceedings of the 1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment, TIDSE 2003, Darmstadt, Germany, March 24-26, 2003*, 2003. [Cited on page 199.]

- Patrick Gebhard, Marc Schröder, Marcela Charfuelan, Christoph Endres, Michael Kipp, Sathish Pammi, Martin Rumpler, and Oytun Türk. IDEAS4Games: Building Expressive Virtual Characters for Computer Games. In Helmut Prendinger, James C. Lester, and Mitsuru Ishizuka, editors, *Proceedings of the 8th International Conference on Intelligent Virtual Agents, IVA 2008, Tokyo, Japan, September 1-3, 2008*, volume 5208 of *Lecture Notes in Computer Science*, pages 426–440. Springer-Verlag Berlin Heidelberg, 2008. [Cited on pages 6, 133, 199, and 201.]
- Patrick Gebhard, Gregor U. Mehlmann, and Michael Kipp. Visual SceneMaker: A Tool for Authoring Interactive Virtual Characters. *Special Issue of the Journal of Multimodal User Interfaces: Interacting with Embodied Conversational Agents*, Springer-Verlag Berlin Heidelberg, 6(1-2):3–11, 2012. [Cited on pages 10, 85, 108, 133, 153, 190, 201, and 239.]
- Patrick Gebhard, Tobias Baur, Ionut Damian, Gregor U. Mehlmann, Johannes Wagner, and Elisabeth André. Exploring Interaction Strategies for Virtual Characters to Induce Stress in Simulated Job Interviews. In Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns, editors, *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2014, Paris, France, May 5-9, 2014*, pages 661–668. International Foundation for Autonomous Agents and Multiagent Systems, 2014. [Cited on pages 151 and 234.]
- Patrick Gebhard, Tanja Schneeberger, Gregor Mehlmann, Tobias Baur, and Elisabeth André. Effects of Real-Time Interruption Handling in Dyadic Interactions with Virtual Social Agents. Submitted to *ACM Transactions on Interactive Intelligent Systems*, 2017. [Cited on pages 151 and 235.]
- Patrick Gebhard. ALMA: A Layered Model of Affect. In Frank Dignum, Virginia Dignum, Sven Koenig, Sarit Kraus, Munindar P. Singh, and Michael Wooldridge, editors, *Proceeding of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2005, Utrecht, The Netherlands, July 25-29, 2005*, pages 29–36. ACM New York, NY, USA, 2005. [Cited on pages 10, 237, and 246.]
- Hartmann J. Genrich. Predicate/Transitions Nets. In K. Jensen and G. Rozenberg, editors, *High-Levels Petri-Nets: Theory and Application*, pages 3–43. Springer Verlag Berlin, 1991. [Cited on pages 87 and 95.]
- Michael P. Georgeff and Francois F. Ingrand. Decision-Making in an Embedded Reasoning System. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence, IJCAI 1989, Detroit, Michigan, August 20 - 25, 1989*. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1989. [Cited on page 83.]
- Dale Gerdemann. Term Encoding of Typed Feature Structures. In *Proceedings of the Fourth International Workshop on Parsing Technologies, Prague, Czech Republic, December, 1995*, pages 89–98, 1995. [Cited on page 120.]
- Darren Gergle, Robert E. Kraut, and Susan R. Fussell. Language Efficiency and Visual Technology: Minimizing Collaborative Effort with Visual Information. *Journal of Language and Social Psychology*, 23(4):491–517, December 2004. [Cited on pages 26 and 27.]
- Darren Gergle, Robert E. Kraut, and Susan R. Fussell. Using Visual Information for Grounding and Awareness in Collaborative Tasks. *Journal of Human-Computer Interaction*, 28(1):1–39, 2013. [Cited on page 28.]
- James J. Gibson and Anne D. Pick. Perception of Another Person's Looking Behavior. *The American Journal of Psychology*, 76(3):386–394, September 1963. [Cited on page 30.]
- Elisa Gironzetti, Lucy Pickering, Meichan Huang, Ying Zhang, Shigehito Menjo, and Salvatore Attardo. Smiling Synchronicity and Gaze Patterns in Dyadic Humorous Conversations. *International Journal of Humor Research*, 29(2):301–324, May 2016. [Cited on page 19.]
- Manuel Giuliani and Alois Knoll. Integrating Multimodal Cues Using Grammar Based Models. In Constantine Stephanidis, editor, *Proceedings of the 4th International Conference on Universal Access in Human-Computer Interaction. Ambient Interaction, UAHCI 2007, Beijing, China, July 22-27, 2007*, pages 858–867. Springer Berlin Heidelberg, 2007. [Cited on page 92.]
- Manuel Giuliani, Michael Kaßecker, Stefan Schwärzler, Alexander Bannat, Jürgen Gast, Frank Wallhoff, Christoph Mayer, Matthias Wimmer, Cornelia Wendt, and Sabrina Schmidt. MuDiS - A Multimodal Dialogue System for Human-Robot Interaction. In *In Proceedings of 1st International Workshop on Cognition for Technical Systems, CoTesys 2008, Munich, Germany, October 6-8, 2008*, 1-6 2008. [Cited on pages 88 and 99.]
- David B. Givens. The nonverbal basis of attraction: flirtation, courtship, and seduction. *Psychiatry*, 41(4):346–359, November 1978. [Cited on page 78.]
- Nadine Glas and Catherine Pelachaud. Definitions of Engagement in Human-Agent Interaction. In *Proceedings of the 6th International Conference on Affective Computing and Intelligent Interaction, ACII 2015, Xian, China, September 21-24, 2015*, pages 944–949, 2015. [Cited on page 76.]

- Arthur M. Glenberg, Jennifer L. Schroeder, and David A. Robertson. Averting the Gaze Disengages the Environment and Facilitates Remembering. *Journal of Memory and Cognition*, 26(4):651–658, July 1998. [Cited on page 37.]
- Stefan Goebel, Ido Aharon Iurgel, Markus Rössler, and Frank Hülsken and Christian Eckes. Design and Narrative Structure for the Virtual Human Scenarios. *The International Journal of Virtual Reality*, 6(4):1–10, 2007. [Cited on page 88.]
- Erving Goffman. *Behavior in Public Places*. The Free Press, Simon and Schuster Inc., New York, NY, USA, 1963. [Cited on pages 34 and 38.]
- Erving Goffman. Footing. *Semiotica*, 25(1-2):1–29, January 1979. [Cited on pages 21, 30, 32, 33, 35, 77, and 183.]
- Julia A. Goldberg. Interrupting the Discourse on Interruptions: An Analysis in Terms of Relationally Neutral, Power- and Rapport-Oriented Acts. *Journal of Pragmatics*, 14(6):883–903, December 1990. [Cited on pages 42, 187, and 195.]
- Charles Goodwin. Restarts, Pauses, and the Achievement of a State of Mutual Gaze at Turn-Beginning. *Sociological Inquiry*, 50(3-4):272–302, July 1980. [Cited on pages 21, 35, 55, and 60.]
- Charles Goodwin. *Conversational Organization: Interaction Between Speakers and Hearers*. New York: Academic Press, New York, NY, USA, 1981. [Cited on pages 21, 35, 36, 55, 60, and 80.]
- Arthur C. Graesser, Shulan Lu, George Tanner Jackson, Heather Hite Mitchell, Mathew Ventura, Andrew Olney, and Max M. Louwerse. AutoTutor: A Tutor with Dialogue in Natural Language. *Behavior Research Methods, Instruments and Computers*, 36(2):180–92, May 2004. [Cited on page 85.]
- Arthur C. Graesser, Patrick Chipman, Brian C. Haynes, and Andrew Olney. AutoTutor: An Intelligent Tutoring System with Mixed-Initiative Dialogue. *IEEE Transactions on Education*, 48(4):612–618, November 2005. [Cited on page 85.]
- Jonathan Gratch, Jeff Rickel, Elisabeth André, Justine Cassell, Eric Petajan, and Norman Badler. Creating Interactive Virtual Humans: Some Assembly Required. *IEEE Intelligent Systems*, 17(4):54–63, July 2002. [Cited on page 10.]
- Jonathan Gratch, Stacy Marsella, Ning Wang, and Brooke Stankovic. Assessing the Validity of Appraisal-Based Models of Emotion. In *Proceedings of the 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops, ACII 2009, Amsterdam, Netherlands, September 10-12, 2009*, pages 1–8, 2009. [Cited on page 10.]
- Zenzi M. Griffin and Kathryn Bock. What Eyes say about Speaking. *Journal of Psychological Science*, 11(4):274–279, July 2000. [Cited on pages 34, 62, 65, and 67.]
- Zenzi M. Griffin. Gaze Durations During Speech Reflect Word Selection and Phonological Encoding. *Journal of Cognition*, 82(1):B1–B14, 2001 2001. [Cited on pages 34, 62, 65, and 67.]
- Barbara J. Grosz and Candace L. Sidner. Attention, Intentions, and the Structure of Discourse. *Journal of Computational Linguistics*, MIT Press Cambridge, MA, USA, 12(3):175–204, July 1986. [Cited on pages 83 and 84.]
- Agneta Gulz. Benefits of Virtual Characters in Computer Based Learning Environments: Claims and Evidence. *International Journal of Artificial Intelligence in Education*, 14(3,4):313–334, December 2004. [Cited on page 229.]
- David L. Hall and James Llinas. An Introduction to Multi-Sensor Data Fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997. [Cited on page 63.]
- Joanna Hall, Terry Tritton, Angela Rowe, Anthony Pipe, Chris Melhuish, and Ute Leonards. Perception of Own and Robot Engagement in Human-Robot Interactions and their Dependence on Robotics Knowledge. *Robotics and Autonomous Systems*, 62(3):392–399, March 2014. [Cited on pages 76 and 99.]
- Stephan Hammer, Birgit Lugin, Sergey Bogomolov, Kathrin Janowski, and Elisabeth André. Investigating Politeness Strategies and Their Persuasiveness for a Robotic Elderly Assistant. In Alexander Meschtscherjakov, Boris De Ruyter, Verena Fuchsberger, Martin Murer, and Manfred Tscheligi, editors, *Proceedings of the 11th International Conference on Persuasive Technology, PERSUASIVE 2016, Salzburg, Austria, April 5-7, 2016*, pages 315–326, Cham, 2016. Springer International Publishing. [Cited on page 238.]

- Amie A. Hane, Stanley Feldstein, and Valerie H. Dernetz. The Relation Between Coordinated Interpersonal Timing and Maternal Sensitivity in Four-Month-Old Infants. *Journal of Psycholinguistic Research*, 32(5):525–539, September 2003. [Cited on page 21.]
- William F. Hanks. *Language and Communicative Practices*. Westview Press, Boulder, CO, USA, 1996. [Cited on page 33.]
- Joy E. Hanna and Susan E. Brennan. Speakers' Eye Gaze Disambiguates Referring Expressions Early During Face-To-Face Conversation. *Journal of Memory and Language*, 57(4):596–615, November 2007. [Cited on pages 5, 27, 29, 34, 56, and 68.]
- Joy E. Hanna, Michael K. Tanenhaus, and John C. Trueswell. The Effects of Common Ground and Perspective on Domains of Referential Interpretation. *Journal of Memory and Language*, 49(1):43–61, 2003. [Cited on pages 27 and 29.]
- John Paulin Hansen, Dan Witzner Hansen, and Anders Sewerin Johansen. Bringing Gaze-Based Interaction Back to Basics. In *Universal Access in HCI, UAHCI 2001, Towards an Information Society for All, Proceedings of the 9th International Conference on Human-Computer Interaction, HCI 2001, New Orleans, USA, August 5-10, 2001*, volume 3, pages 325–329. Lawrence Erlbaum, Mahwah, NJ, USA, 2001. [Cited on pages 66 and 159.]
- David Harel and Hillel Kugler. The Rhapsody Semantics of Statecharts (or, On the Executable Core of the UML). In Hartmut Ehrig, Werner Damm, Jörg Desel, Martin Große-Rhode, Wolfgang Reif, Eckehard Schnieder, and Engelbert Westkämper, editors, *Integration of Software Specification Techniques for Applications in Engineering, Priority Program SoftSpez of the German Research Foundation (DFG), Final Report*, volume 3147 of *Lecture Notes in Computer Science*, pages 325–354. Springer Berlin Heidelberg, 2004. [Cited on pages 56, 86, 201, and 205.]
- David Harel and Amnon Naamad. The STATEMATE Semantics of Statecharts. *ACM Transactions on Software Engineering and Methodology*, 5(4):193–333, October 1996. [Cited on pages 56, 86, 201, and 205.]
- David Harel and Michal Politi. *Modeling Reactive Systems with Statecharts: The Statemate Approach*. McGraw-Hill, Inc. New York, NY, USA, 1998. [Cited on pages 10, 51, 56, 59, 86, 142, 147, and 205.]
- David Harel, Hagi Lachover, Amnon Naamad, Amir Pnueli, Michal Politi, Rivi Sherman, Aharon Shtull-Trauring, and Mark Trakhtenbrot. STATEMATE: A Working Environment for the Development of Complex Reactive Systems. *IEEE Transactions on Software Engineering*, 16(4):403–414, April 1990. [Cited on pages 56, 86, and 205.]
- David Harel. Statecharts: A Visual Formalism for Complex Systems. *Science of Computer Programming*, 8(3):231–274, June 1987. [Cited on pages 10, 51, 56, 59, 86, 108, 147, and 205.]
- Jinni A. Harrigan. Listener's Body Movements and Speaking Turns. *Communication Research*, 12(2):233–250, April 1985. [Cited on page 35.]
- Donald P. Hayes and Loren Cobb. Cycles of Spontaneous Conversation under Longterm Isolation. In M. Davis, editor, *Interaction Rhythms: Periodicity in Communicative Behavior*, pages 319–340. Human Science Press, New York, NY, USA, 1982. [Cited on pages 20 and 59.]
- Frederick Hayes-Roth. Rule-Based Systems. *Communications of the ACM*, 28(9):921–932, 1985. [Cited on page 82.]
- Marcel Heerink, Ben Kröse, Vanessa Evers, and Bob Wielinga. The Influence of Social Presence on Acceptance of a Companion Robot by Older People. *Journal of Physical Agents*, 2(2):33–40, 2008. [Cited on page 6.]
- Marcel Heerink, Ben Kröse, Vanessa Evers, and Bob Wielinga. Influence of Social Presence on Acceptance of an Assistive Social Robot and Screen Agent by Elderly Users. *Journal of Advanced Robotics, Special Issue on Robot and Human Interactive Communication*, 23(14):1909–1923, 2009. [Cited on page 6.]
- Alexis Heloir and Michael Kipp. EMBR: A Realtime Animation Engine for Interactive Embodied Agents. In *Proceedings of the 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops, ACII 2009*,, pages 1–2. IEEE, 2009. [Cited on page 97.]
- Alexis Heloir and Michael Kipp. Realtime animation of interactive agents: Specification and realization. *Journal of Applied Artificial Intelligence*, 24(6):510–529, 2010. [Cited on pages 97 and 99.]
- Felienne Hermans, Martin Pinzger, and Arie Van Deursen. Domain-Specific Languages in Practice: A User Study on the Success Factors. In *Proceedings of the 12th International Conference on Model Driven Engineering Languages and Systems, MODELS 2009, Denver, CO, USA, October 3 - 9, 2009*, volume 5795 of *Lecture Notes in Computer Science*, pages 423–437. Springer Berlin Heidelberg, 2009. [Cited on page 104.]

- Gerd Herzog and Norbert Reithinger. The SmartKom Architecture: A Framework for Multimodal Dialogue Systems. In Wolfgang Wahlster, editor, *SmartKom: Foundations of Multimodal Dialogue Systems*, pages 55–70. Springer Berlin Heidelberg, 2006. [Cited on page 83.]
- Gerd Herzog, Alassane Ndiaye, Stefan Merten, Heinz Kirchmann, Tilman Becker, and Peter Poller. Large-Scale Software Integration for Spoken Language and Multimodal Dialog Systems. *Natural Language Engineering, Cambridge University Press*, 10(3/4):283–305, October 2004. [Cited on pages 83 and 90.]
- Ursula Hess and Sylvie Blairy. Facial Mimicry and Emotional Contagion to Dynamic Emotional Facial Expressions and their Influence on Decoding Accuracy. *International Journal of Psychophysiology*, 40(2):129–141, March 2001. [Cited on page 18.]
- Ursula Hess and Agneta Fischer. Emotional Mimicry as Social Regulation. *Personality and Social Psychology Review*, 17(2):142–157, May 2013. [Cited on pages 18, 19, 37, and 182.]
- Ursula Hess and Agneta Fischer. Emotional Mimicry: Why and When We Mimic Emotions. *Social and Personality Psychology Compass*, 8(2):45–57, February 2014. [Cited on pages 19 and 37.]
- Dirk Heylen, Stefan Kopp, Stacy C. Marsella, Catherine Pelachaud, and Hannes Vilhjálmsón. The Next Step Towards a Function Markup Language. In Helmut Prendinger, James C. Lester, and Mitsuru Ishizuka, editors, *Proceedings of the 8th International Conference on Intelligent Virtual Agents, IVA 2008, Tokyo, Japan, September 1-3, 2008*, volume 5208 of *Lecture Notes in Computer Science*, pages 270–280. Springer-Verlag Berlin Heidelberg, 2008. [Cited on pages 97, 98, and 109.]
- Graeme Hirst, Susan McRoy, Peter Heeman, Philip Edmonds, and Diane Horton. Repairing Conversational Misunderstandings and Non-Understandings. *Journal of Speech Communication*, 15:213–230, 1994. [Cited on pages 59 and 70.]
- Anna Hjalmarsson and Catharine Oertel. Gaze Direction as a Back-Channel Inviting Cue in Dialogue. In *Proceedings of the 12th International Conference on Intelligent Virtual Agent, IVA 2012, Workshop on Real-Time Conversations with Virtual Agents, RCVA 2012, Santa Cruz, California, USA, September 15th, 2012*, 2012. [Cited on pages 36 and 181.]
- Anna Hjalmarsson, Preben Wik, and Jenny Brusk. Dealing with DEAL: A Dialogue System for Conversation Training. In *In Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, Antwerp, September 2007*, pages 132–135. Association for Computational Linguistics, 2007. [Cited on page 86.]
- Simon Ho, Tom Foulsham, and Alan Kingstone. Speaking and Listening with the Eyes: Gaze Signaling during Dyadic Interactions. *PLoS ONE*, 10(8), August 2015. [Cited on page 35.]
- James G. Hollandsworth, JR. Richard Kazelskis, Joanne Stevens, and Mary Edith Dressel. Relative Contributions of Verbal, Articulative, and Nonverbal Communication to Employment Decisions in the Job Interview Setting. *Journal of Personnel Psychology*, 32(2):359–367, Summer 1979. [Cited on page 233.]
- Aaron Holroyd and Charles Rich. Using the Behavior Markup Language for Human-Robot Interaction. In *Proceedings of the 7th ACM/IEEE International Conference on Human-Robot Interaction, HRI 2012, Boston, Massachusetts, USA, March 5-8, 2012*, pages 147–148, 2012. [Cited on page 98.]
- Aaron Holroyd, Charles Rich, Candace L. Sidner, and Brett Ponsler. Generating Connection Events for Human-Robot Collaboration. In *Proceedings of the 20th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN 2011, Atlanta, GA, 31 July - 3 August, 2011*, pages 241–246, 2011. [Cited on pages 55, 76, 87, 99, 176, 177, and 180.]
- Hartwig Holzapfel, Kai Nickel, and Rainer Stiefelwagen. 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 2004, State College, PA, USA, October 13 - 15, 2004*, pages 175–182. ACM, New York, NY, USA, 2004. [Cited on pages 92, 93, and 119.]
- Lode Hoste, Bruno Dumas, and Beat Signer. Mudra: A Unified Multimodal Interaction Framework. In *Proceedings of the 13th International Conference on Multimodal Interfaces, ICMI 2011, Alicante, Spain, November 14-18, 2011*, pages 97–104. ACM, New York, NY, USA, 2011. [Cited on pages 64, 82, 93, 99, and 118.]
- Julian Hough and David Schlangen. Investigating fluidity for human-robot interaction with real-time, real-world grounding strategies. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGDIAL 2016, Los Angeles, CA, USA, September, 13-25, 2016*, pages 288–298, 2016. [Cited on page 54.]

- Michael J. Hove and Jane L. Risen. It's All in the Timing: Interpersonal Synchrony Increases Affiliation. *Journal on Social Cognition*, 27(6):949–961, 2009. [Cited on pages 17, 21, and 29.]
- Chien-Ming Huang and Bilge Mutlu. Robot Behavior Toolkit: Generating Effective Social Behaviors for Robots. In *In Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction, HRI 2012, Boston, MA, USA, March 5-8, 2012*, pages 25–32, 2012. [Cited on pages 74 and 75.]
- Chien-Ming Huang and Andrea L. Thomaz. Joint Attention in Human-Robot Interaction. In *Dialog with Robots: Papers from the AAAI Fall Symposium*, 2010. [Cited on pages 74 and 76.]
- Chien-Ming Huang and Andrea L. Thomaz. Effects of Responding to, Initiating and Ensuring Joint Attention in Human-Robot Interaction. In *Proceedings of the 20th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN 2011, Atlanta, GA, 31 July - 3 August, 2011*, pages 65 – 71, 2011. [Cited on pages 74, 75, 76, and 99.]
- Chien-Ming Huang, Sean Andrist, Allison Saupé, and Bilge Mutlu. Using Gaze Patterns to Predict Task Intent in Collaboration. *Journal of Frontiers in Psychology*, 6(1049):1–12, July 2015. [Cited on pages 5, 33, 54, 56, and 74.]
- Scott E. Hudson and Gary L. Newell. Probabilistic State Machines: Dialog Management for Inputs with Uncertainty. In *Proceedings of the 5th Annual ACM Symposium on User Interface Software and Technology, UIST 1992, Monterey, California, USA, November 15 - 18, 1992*, pages 199–208. ACM, New York, NY, USA, 1992. [Cited on page 95.]
- Dell Hymes. Models of the Interaction of Language and Social Life. In J. Gumperz and D. Hymes, editors, *Directions in Sociolinguistics: The Ethnography of Communication*, pages 35–71. Holt, Rhinehart and Winston, New York, NY, USA, 1972. [Cited on page 33.]
- Michita Imai, Tetsuo Ono, and Hiroshi Ishiguro. Physical Relation and Expression: Joint Attention for Human-Robot Interaction. *IEEE Transactions on Industrial Electronics*, 50(4):636–643, August 2003. [Cited on page 74.]
- Ido A. Iurgel, Rogério E. da Silva, Pedro R. Ribeiro, Abel B. Soares, and Manuel Filipe dos Santos. CREAATOR – An Authoring Framework for Virtual Actors. In Zsófia Ruttkay, Michael Kipp, Anton Nijholt, and Hannes Högni Vilhjálmsson, editors, *Proceedings of the 9th International Conference on Intelligent Virtual Agents, IVA 2009, Amsterdam, The Netherlands, September 14-16, 2009*, pages 562–563. Springer-Verlag, Berlin, Heidelberg, 2009. [Cited on page 82.]
- Ido Iurgel. From Another Point of View: Art-E-Fact. In Stefan Göbel, Ulrike Spierling, Anja Hoffmann, Ido Iurgel, Oliver Schneider, Johanna Dechau, and Axel Feix, editors, *Proceedings of the 2nd 2nd International Conference on Technologies for Interactive Digital Storytelling and Entertainment, TIDSE 2004, Darmstadt, Germany, June 24-26, 2004 Proceedings*, pages 26–35. Springer Berlin Heidelberg, 2004. [Cited on pages 87 and 88.]
- Ido Iurgel. Cyranus - An Authoring Tool for Interactive Edutainment Applications. In Zhigeng Pan, Ruth Aylett, Holger Diener, Xiaogang Jin, Stefan Göbel, and Li Li, editors, *Proceedings of the 1st International Conference on Technologies for E-Learning and Digital Entertainment, Edutainment 2006, Hangzhou, China, April 16-19, 2006*, volume 3942 of *Lecture Notes in Computer Science*, pages 577–580. Springer-Verlag, Berlin, Heidelberg, 2006. [Cited on pages 87 and 133.]
- Peter H.M. Jacobs and Alexander Verbraeck. Single-Threaded Specification of Process-Interaction Formalism in Java. In *Proceedings of the 36th Conference on Winter Simulation, WSC 2004, Washington, DC, USA, December 5-8, 2004*, volume 2, pages 1548–1555. IEEE, 2004. [Cited on page 218.]
- Joseph Jaffe and Stanley Feldstein. *Rhythms of Dialogue (Personality and Psychopathology)*. Academic Press Inc, 1979. [Cited on page 40.]
- Alejandro Jaimes and Nicu Sebe. Multimodal Human-Computer Interaction: A Survey. *Computer Vision and Image Understanding: Special Issue on Vision for Human-Computer Interaction*, 108(1-2):116–134, October-November 2007. [Cited on pages 56, 62, 65, and 68.]
- Deborah James and Sandra Clarke. Women, Men, and Interruptions: A Critical Review. In Deborah Tannen, editor, *Gender and Conversational Interaction*, pages 231–280. New York: Oxford University Press, Inc, 1993. [Cited on page 40.]
- Michael Jasnow, Cynthia L. Crown, Stanley Feldstein, Linda Taylor, Beatrice Beebe, and Joseph Jaffe. Coordinated Interpersonal Timing of Down-Syndrome and Nondelayed Infants with Their Mothers: Evidence for a Buffered Mechanism of Social Interaction. *Biological Bulletin*, 175:355–360, December 1988. [Cited on page 21.]

- Gail Jefferson. A Case of Precision Timing in Ordinary Conversation: Overlapped Tag-Positioned Address Terms in Closing Sequences. *Semiotica*, 9(1):47–96, 1973. [Cited on page 76.]
- W. Lewis Johnson, Jeff W. Rickel, and James C. Lester. Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments. *International Journal of Artificial Intelligence in Education*, 11:47–78, 2000. [Cited on pages 87 and 229.]
- Michael Johnston and Srinivas Bangalore. Finite-State Multimodal Parsing and Understanding. In *Proceedings of the 18th Conference on Computational Linguistics, COLING 2000, Saarbrücken, Germany, July 31 - August 04, 2000*. Association for Computational Linguistics, Stroudsburg, PA, USA, 2000. [Cited on pages 92 and 94.]
- Michael Johnston and Srinivas Bangalore. Finite-State Methods for Multimodal Parsing and Integration. In *Proceedings of the Workshop on Finite State Methods in Natural Language Processing held on the 13th European Summer School in Logic, Language and Information, ESSLI 2001, Helsinki, Finland, August 20-24, 2001, 2001*. [Cited on pages 94 and 118.]
- Michael Johnston and Srinivas Bangalore. MATCHkiosk: A Multimodal Interactive City Guide. In *The Companion Volume to the Proceedings of 42st Annual Meeting of the Association for Computational Linguistics, ACL 2004, Barcelona, Spain, July 21-26, 2004*, pages 222–225, 2004. [Cited on page 94.]
- Michael Johnston and Srinivas Bangalore. Finite-State Multimodal Integration and Understanding. *Journal of Natural Language Engineering*, 11(2):159–187, June 2005. [Cited on pages 94, 118, and 133.]
- Michael Johnston, Philip R. Cohen, David McGee, Sharon L. Oviatt, James A. Pittman, and Ira Smith. Unification-Based Multimodal Integration. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics, Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain, July 7-12, 1997*, pages 281–288. Morgan Kaufmann, San Francisco, CA, USA, 1997. [Cited on pages 91, 99, 118, and 119.]
- Michael Johnston, Srinivas Bangalore, Gunaranjan Vasireddy, Amanda Stent, Patrick Ehlen, Marilyn Walker, Steve Whittaker, and Preetam Maloor. MATCH: An Architecture for Multimodal Dialogue System. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, ACL 2002, Philadelphia, Pennsylvania, USA, July 6-12, 2002*, pages 376–383, 2002. [Cited on page 94.]
- Michael Johnston. Multimodal Language Processing. In *Proceedings of the 5th International Conference on Spoken Language Processing, ICSLP 1998, Incorporating the 7th Australian International Speech Science and Technology Conference, Sydney, Australia, November 30 - December 4, 1998, 1998*. [Cited on page 92.]
- Michael Johnston. Unification-Based Multimodal Parsing. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics, COLING 1998, and 17th International Conference on Computational Linguistics Montreal, ACL 1998, Montreal, Quebec, Canada, August 10-14, 1998*, pages 624–630. Association for Computational Linguistics, Stroudsburg, PA, USA, 1998. [Cited on pages 92, 93, 99, 118, and 119.]
- Michael Johnston. Deixis and Conjunction in Multimodal Systems. In *Proceedings of the 18th Conference on Computational Linguistics, COLING 2000, Saarbrücken, Germany, July 31 - August 4, 2000*. Association for Computational Linguistics, 2000. [Cited on pages 93 and 118.]
- Michael Johnston. Building Multimodal Applications with EMMA. In James L. Crowley, Yuri A. Ivanov, Christopher Richard Wren, Daniel Gatica-Perez, Michael Johnston, and Rainer Stiefelwagen, editors, *Proceedings of the 11th International Conference on Multimodal Interfaces and the Sixth Workshop on Machine Learning for Multimodal Interaction, ICMI-MLMI 2009, Cambridge, Massachusetts, USA, November 2-6, 2009*, pages 47–54. ACM, New York, NY, USA, 2009. [Cited on pages 64, 90, 119, and 125.]
- Kristiina Jokinen, Hirohisa Furukawa, Masafumi Nishida, and Seiichi Yamamoto. Gaze and Turn-taking Behavior in Casual Conversational Interactions. *ACM Transactions on Interactive Intelligent Systems (TiiS) - Special issue on interaction with smart objects, Special section on eye gaze and conversation*, 3(2):12:1–12:30, August 2013. [Cited on pages 15 and 30.]
- Yvonne Jung, Arjan Kuijper, Dieter W. Fellner, Michael Kipp, Jan Miksatko, Jonathan Gratch, and Daniel Thalmann. Believable Virtual Characters in Human-Computer Dialogs. In W. John and B. Wyvill, editors, *Proceedings of the 32nd Annual Conference of the European Association for Computer Graphics, Eurographics 2011, Llandudno, UK, April 11 - 15, 2011*, pages 75–100, 2011. [Cited on page 81.]

- Edward C. Kaiser and Philip R. Cohen. Implementation Testing of a Hybrid Symbolic/Statistical Multimodal Architecture. In John H. L. Hansen and Bryan L. Pellom, editors, *Proceedings of the 7th International Conference on Spoken Language Processing, ICSLP 2002 - INTERSPEECH 2002, Denver, Colorado, USA, September 16-20, 2002, 2002*. [Cited on page 92.]
- Ed Kaiser, Alex Olwal, David McGee, Hrvoje Benko, Andrea Corradini, Xiaoguang Li, Phil Cohen, and Steven Feiner. Mutual Disambiguation of 3D Multimodal Interaction in Augmented and Virtual Reality. In Sharon L. Oviatt, Trevor Darrell, Mark T. Maybury, and Wolfgang Wahlster., editors, *Proceedings of the 5th International Conference on Multimodal Interfaces, ICMI 2003, Vancouver, British Columbia, Canada, November 5-7, 2003, 2003*. [Cited on pages 56, 65, 92, and 119.]
- Ronald M. Kaplan and Joan Bresnan. Lexical-Functional Grammar: A Formal System for Grammatical Representation. In Joan Bresnan, editor, *The Mental Representation of Grammatical Relations*, pages 173–281. The MIT Press, Cambridge, MA, USA, 1982. [Cited on page 119.]
- Frédéric Kaplan and Verena V. Hafner. The Challenges of Joint Attention. *Journal of Interaction Studies*, 7(2):135–169, 2006. [Cited on pages 31 and 75.]
- Leonard Karakowsky, Kenneth McBey, and Diane L. Miller. Gender, perceived competence, and power displays: Examining verbal interruptions in a group context. *Journal of Small Group Research*, 35(4):407–439, August 2004. [Cited on pages 5 and 44.]
- Robert T. Kasper and William C. Rounds. A Logical Semantics for Feature Structures. In *Proceedings of the 24th Annual Meeting on Association for Computational Linguistics, ACL 1986, New York, NY, USA, July 10-13, 1986*, pages 257–266. Association for Computational Linguistics, Stroudsburg, PA, USA, 1986. [Cited on pages 11, 52, 91, and 119.]
- Robert T. Kasper and William C. Rounds. The Logic of Unification in Grammar. *Journal of Linguistics and Philosophy, Kluwer Academic Publishers*, 13(1):35–58, February 1990. [Cited on pages 63, 91, and 119.]
- Manpreet Kaur, Marilyn Tremaine, Ning Huang, Joseph Wilder, Zoran Gacovski, Frans Flippo, and Chandra Sekhar Mantravadi. Where is "It"? Event Synchronization in Gaze-Speech Input Systems. In Sharon L. Oviatt, Trevor Darrell, Mark T. Maybury, and Wolfgang Wahlster., editors, *Proceedings of the 5th International Conference on Multimodal Interfaces, ICMI 2003, Vancouver, British Columbia, Canada, November 5-7, 2003*, pages 151–158. ACM, New York, NY, USA, 2003. [Cited on pages 35, 56, 62, 65, and 67.]
- Martin Kay. Functional Grammar. In *Proceedings of the 5th Annual Meeting of the Berkley Linguistics Society, Berkley, CA, USA, 1979*, pages 142–158. Berkley Linguistics Society, Berkley, CA, USA, 1979. [Cited on pages 91 and 119.]
- Martin Kay. Functional Unification Grammar: A Formalism for Machine Translation. In *Proceedings of the 10th International Conference on Computational Linguistics and the 22nd Annual Meeting on Association for Computational Linguistics, ACL 1984, Stanford, CA, USA, July 5, 1984*, pages 75–78. Association for Computational Linguistics, Stroudsburg, PA, USA, 1984. [Cited on page 119.]
- Dennis K. Keenan and Dag Westerståhl. Generalized Quantifiers in Linguistics and Logic. In Johan Van Benthem and Alice Ter Meulen, editors, *Handbook of Logic and Language*, pages 837–893. Elsevier, 2011. [Cited on page 130.]
- Adam Kendon and Mark Cook. The Consistency of Gaze Patterns in Social Interaction. *British Journal of Psychology*, 60:481–494, 1969. [Cited on page 38.]
- Adam Kendon. Some Functions of Gaze-Direction in Social Interaction. *Acta Psychologica, North-Holland Publishing Co., Amsterdam, The Netherlands*, 26(1):22–63, 1967. [Cited on pages 5, 15, 17, 21, 33, 35, 36, 37, 38, 42, 43, 54, 55, 56, 68, 77, 179, 181, and 235.]
- Adam Kendon. Movement and Coordination in Social Interaction: Some Examples Described. *Acta Psychologica*, 32(2):100–125, April 1970. [Cited on pages 17 and 18.]
- Carol W. Kennedy and Carl T. Camden. A New Look at Interruptions. *Western Journal of Speech Communication*, 47(1):45–58, 1983. [Cited on pages 5, 41, 43, and 44.]
- Carol W. Kennedy and Carl T. Camden. Interruptions and Nonverbal Gender Differences. *Journal of Nonverbal Behavior*, 8(2):91–108, December 1983. [Cited on pages 5, 40, and 44.]

- Casey Kennington, Livia Dia, and David Schlangen. A Discriminative Model for Perceptually-Grounded Incremental Reference Resolution. In *Proceedings of the 11th International Conference on Computational Semantics, IWCS 2015, London, UK, April 15-17 2015*, pages 195–205, 2015. [Cited on pages 68 and 70.]
- Patrick Kenny, Thomas D. Parsons, Jonathan Gratch, Anton Leuski, and Albert A. Rizzo. Virtual Patients for Clinical Therapist Skills Training. In Catherine Pelachaud, Jean-Claude Martin, Elisabeth André, Gérard Chollet, Kostas Karpouzis, and Danielle Pelé, editors, *Proceedings of the 7th International Conference on Intelligent Virtual Agents, IVA 2007, Paris, France, September 17-19, 2007*, volume 4722 of *Lecture Notes in Computer Science*, pages 197–210, 2007. [Cited on page 6.]
- Michael Kipp, Michael Neff, Kerstin H. Kipp, and Irene Albrecht. Towards Natural Gesture Synthesis: Evaluating Gesture Units in a Data-Driven Approach to Gesture Synthesis. In *Proceedings of the 7th International Conference on Intelligent Virtual Agents, IVA 2007, Paris, France, September 17-19, 2007*, pages 15–28, 2007. [Cited on page 10.]
- Michael Kipp, Alexis Heloir, Marc Schröder, and Patrick Gebhard. Realizing Multimodal Behavior: Closing the Gap between Behavior Planning and Embodied Agent Presentation. In Jan M. Allbeck, Norman I. Badler, Timothy W. Bickmore, Catherine Pelachaud, and Alla Safonova, editors, *Proceedings of the 10th International Conference on Intelligent Virtual Agents, IVA 2010, Philadelphia, Pennsylvania, USA, September 20-22, 2010*, volume 6356 of *Lecture Notes in Computer Science*, pages 57–63. Springer-Verlag Berlin Heidelberg, 2010. [Cited on pages 96, 97, 98, and 109.]
- Michael Kipp. Anvil - A Generic Annotation Tool for Multimodal Dialogue. In Paul Dalsgaard, Boslarsgr Lindberg, Henrik Benner, and Zheng-Hua Tan, editors, *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech)*, pp. 1367-1370., pages 1367–1370. ISCA, 2001. [Cited on page 245.]
- Michael Kipp. ANVIL: A Universal Video Research Tool. In Jacques Durand, Ulrike Gut, and Gjert Kristofferson, editors, *Handbook of Corpus Phonology*, chapter 21, pages 420–436. Oxford University Press, 2014. [Cited on page 245.]
- Felix Kistler. *Full Body Interaction - Design, Implementation, and User Support*. PhD thesis, Augsburg University, Faculty of Applied Computer Science, 2016. [Cited on pages 153 and 154.]
- Robert E. Kleck and William Nuessle. Congruence between the Indicative and Communicative Functions of Eye-Contact in Interpersonal Relations. *British Journal of Social and Clinical Psychology*, 7(4):241–246, December 1968. [Cited on page 38.]
- Chris L. Kleinke. Gaze and Eye Contact: A Research Review. *Psychological Bulletin*, 100(1):78–100, July 1986. [Cited on pages 15, 34, and 35.]
- Martin Klesen, Michael Kipp, Patrick Gebhard, and Thomas Rist. Staging Exhibitions: Methods and Tools for Modelling Narrative Structure to Produce Interactive Performances with Virtual Actors. *Journal of Virtual Reality, Springer-Verlag, London*, 7(1):17–29, December 2003. [Cited on page 199.]
- Christoph Klimmt, Christian Roth, Ivar Vermeulen, Peter Vorderer, and Franziska Susanne Roth. Forecasting the Experience of Future Entertainment Technology: Interactive Storytelling and Media Enjoyment. *Journal of Games and Culture*, 7(3):187–208, May 2012. [Cited on page 231.]
- Mark L. Knapp, Judith A. Hall, and Terrence G. Horgan. *Non-Verbal Communication in Human Interaction*. Wadsworth Cengage Learning, 2014. [Cited on page 56.]
- Alfred Kobsa, Jürgen Allgayer, Carola Reddig, Norbert Reithinger, Dagmar Schmauks, Karin Harbusch, and Wolfgang Wahlster. Combining Deictic Gestures and Natural Language for Referent Identification. In *Proceedings of the 11th Conference on Computational Linguistics, COLING 1986, Bonn, Germany, August 25 - 29, 1986*, pages 356–361. Association for Computational Linguistics Stroudsburg, PA, USA, 1986. [Cited on page 89.]
- Alexander Koller and Geert-Jan M. Kruiff. Talking Robots with Lego Mindstorms. In *Proceedings of the 20th International Conference on Computational Linguistics, COLING 2004, Geneva, Switzerland, August 23-27, 2004*, pages 336–342, 2004. [Cited on page 85.]
- David B. Koons, Carlton J. Sparrell, and Kristinn R. Thorisson. Integrating Simultaneous Input from Speech, Gaze, and Hand Gestures. In Mark T. Maybury, editor, *Intelligent Multimedia Interfaces*, pages 257–276. American Association for Artificial Intelligence Menlo Park, CA, USA, 1993. [Cited on page 90.]
- Stephan Kopp and Ipke Wachsmuth. Synthesizing Multimodal Utterances for Conversational Agents. *Computer Animation and Virtual Worlds*, 15(1):39–52, 2004. [Cited on pages 97 and 99.]

- Stephan Kopp, Bernhard Jung, Nadine Leßmann, and Ipke Wachsmuth. Max - A Multimodal Assistant in Virtual Reality Construction. *KI - Künstliche Intelligenz*, 4(3):11–17, 2003. [Cited on page 97.]
- Stefan Kopp, Brigitte Krenn, Stacy Marsella, Andrew N. Marshall, Catherine Pelachaud, Hannes Pirker, Kristinn R. Thórisson, and Hannes Vilhjálmsón. Towards a Common Framework for Multimodal Generation: The Behavior Markup Language. In Jonathan Gratch, Michael Young, Ruth Aylett, Daniel Ballin, and Patrick Olivier, editors, *Proceedings of the 6th International Conference on Intelligent Virtual Agents, IVA 2006, Marina Del Rey, CA, USA, August 21-23, 2006*, volume 4133 of *Lecture Notes in Computer Science*, pages 205–217. Springer-Verlag Berlin Heidelberg, 2006. [Cited on pages 88, 96, 97, 98, 99, and 109.]
- Stefan Kopp, Herwin van Welbergen, Ramin Yaghoubzadeh, and Hendrik Buschmeier. An Architecture for Fluid Real-Time Conversational Agents: Integrating Incremental Output Generation and Input Processing. *Journal on Multimodal User Interfaces*, 8:97–108, 2014. [Cited on pages 88 and 99.]
- Tomaž Kosar and Marjan Mernik. Embedded Domain-Specific Languages in Prolog. *Acta Electrotechnica et Informatica*, 6(3):1–6, January 2006. [Cited on page 67.]
- Timothy Koschmann and Curtis D. LeBaron. Reconsidering Common Ground: Examining Clark’s Contribution Theory in the OR. In *Proceedings of the 8th European Conference on Computer Supported Cooperative Work, ECSCW 2003, Helsinki, Finland, September 14-18, 2003*, pages 81–98, 2003. [Cited on page 22.]
- Robert A. Kowalski. Predicate Logic as Programming Language. In *Proceedings of IFIP Congress 1974*, pages 569–574. North-Holland Publishing, Amsterdam, The Netherlands, 1974. [Cited on pages 65, 67, 108, and 121.]
- Robert A. Kowalski. Algorithm = Logic + Control. *Communications of the ACM*, 22(7):424–436, July 1979. [Cited on pages 65, 67, 108, and 121.]
- Brigitte Krenn and Hannes Pirker. Defining the Gesticon: Language and Gesture Coordination for Interacting Embodied Agents. In *In Proceedings of the Symposium on Language, Speech and Gesture for Expressive Characters, AISB 2004, Leeds, UK, 2004*, pages 107–115, 2004. [Cited on page 97.]
- Fredrik Kronlid and Torbjörn Lager. Implementing the Information-State Update Approach to Dialogue Management in a Slightly Extended SCXML. In *Proceedings of the 11th International Workshop on the Semantics and Pragmatics of Dialogue, DECALOG 2007, Trento, Italy, May 30 - June 1, 2007*, 2007. [Cited on pages 86 and 133.]
- Fredrik Kronlid. Turn Taking for Artificial Conversational Agents. In Matthias Klusch, Michael Rovatsos, and Terry R. Payne, editors, *Proceedings of the 10th International Conference on Cooperative Information Agents, CIA 2006, Edinburgh, UK September 11-13, 2006*, volume 4149 of *Lecture Notes on Artificial Intelligence*, pages 81–95. Springer-Verlag Berlin, Heidelberg, 2006. [Cited on page 86.]
- Fredrik Kronlid. *Steps Towards Multi-Party Dialogue Management*. PhD thesis, The Graduate School of Language Technology, Department of Linguistics, University of Gothenburg, Göteborg, Sweden, 2008. [Cited on page 86.]
- Wojciech Marek Kulesza, Aleksandra Cislak, Robin R. Vallacher, Andrzej Nowak, Martyna Czekiel, and Sylwia Bedynska. The face of the chameleon: The experience of facial mimicry for the mimicker and the mimickee. *Journal of Social Psychology*, 155(6):590–604, 2015. [Cited on page 19.]
- Ashwani Kumar and Laurent Romary. Comprehensive Framework for MultiModal Meaning Representation. In *Proceedings of the 5th International Workshop on Computational Semantics, IWCS 2003, Tilburg, Netherlands, January, 2003*, pages 225–251, 2003. [Cited on page 90.]
- Yoshinori Kuno, Kazuhisa Sadazuka, Michie Kawashima, Sachie Tsuruta, Keiichi Yamazaki, and Akiko Yamazaki. Effective Head Gestures for Museum Guide Robots in Interaction with Humans. In *Proceedings of the 16th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN 2007, Jeju Island, Korea, August 26-29, 2007*, pages 151–156, 2007. [Cited on page 76.]
- Yoshinori Kuno, Kazuhisa Sadazuka, Michie Kawashima, Keiichi Yamazaki, Akiko Yamazaki, and Hideaki Kuzuoka. Museum guide robot based on sociological interaction analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2007, San Jose, California, USA, April 28 - May 3, 2007*, pages 1191–1194, ACM, New York, NY, USA, 2007. [Cited on page 76.]
- Jean-François Ladry, David Navarre, and Philippe Palanque. Formal Description Techniques to Support the Design, Construction and Evaluation of Fusion Engines for SURE (Safe, Usable, Reliable and Evolvable) Multimodal Interfaces. In James L. Crowley, Yuri A. Ivanov, Christopher Richard Wren, Daniel Gatica-Perez, Michael Johnston, and Rainer Stiefelhagen, editors, *Proceedings of the 11th International Conference on Multimodal Interfaces and the Sixth Workshop on Machine Learning for Multimodal Interaction, ICMI-MLMI 2009*,

- Cambridge, Massachusetts, USA, November 2-6, 2009, pages 185–192. ACM, New York, NY, USA, 2009. [Cited on page 95.]
- Marianne LaFrance. Nonverbal Synchrony and Rapport: Analysis by the Cross-Lag Panel Technique. *Journal of Social Psychology Quarterly*, 42(1):66–70, March 1979. [Cited on page 19.]
- Marianne LaFrance. Interaction Rhythms: Periodicity in Communicative Behavior. In Martha Davis, editor, *Posture Mirroring and Rapport*, pages 279–98. Human Sciences Press, New York, NY, USA, 1982. [Cited on pages 18 and 19.]
- Jessica L. Lakin and Tanya L. Chartrand. Using Nonconscious Behavioral Mimicry to Create Affiliation and Rapport. *Journal of Psychological Science*, 14(4):334–339, July 2003. [Cited on pages 18, 19, and 29.]
- Jessica L. Lakin, Valerie E. Jefferis, Clara Michelle Cheng, and Tanya L. Chartrand. The Chameleon Effect as Social Glue: Evidence for the Evolutionary Significance of Nonconscious Mimicry. *Journal of Nonverbal Behavior*, 27(3):145–162, September 2003. [Cited on pages 18 and 28.]
- Jessica L. Lakin. Behavioral Mimicry and Interactional Synchrony. In Judith L. Hall and Mark L. Knapp, editors, *Handbook of Communication Science*, chapter 18, pages 539–. De Gruyter Mouton, 2012. [Cited on pages 17 and 18.]
- Denis Lalanne, Laurence Nigay, Philippe Palanque, Peter Robinson, Jean Vanderdonckt, and Jean-François Ladry. Fusion Engines for Multimodal Input: A Survey. In James L. Crowley, Yuri A. Ivanov, Christopher Richard Wren, Daniel Gatica-Perez, Michael Johnston, and Rainer Stiefelwagen, editors, *Proceedings of the 11th International Conference on Multimodal Interfaces and the Sixth Workshop on Machine Learning for Multimodal Interaction, ICMI-MLMI 2009, Cambridge, Massachusetts, USA, November 2-6, 2009*, pages 153–160. ACM, New York, NY, USA, 2009. [Cited on pages 62 and 89.]
- Kathrin Lambertz. Back-Channelling: The Use of Yeah and Mm to Portray Engaged Listenership. *Griffiths Working Papers in Pragmatics and Intercultural Communication*, 4(1-2):11–18, 2011. [Cited on pages 25 and 179.]
- Leslie Lamport. Time, Clocks, and the Ordering of Events in a Distributed System. *Communications of the ACM*, 21(7):558–565, July 1978. [Cited on page 145.]
- Leslie Lamport. On Interprocess Communication: Part I - Basic Formalism. *Journal of Distributed Computing*, 1(2):77–85, 1986. [Cited on pages 58 and 144.]
- Leslie Lamport. The Mutual Exclusion Problem: Part I - A Theory of Interprocess Communication. *Journal of the ACM*, 33(2):313–326, April 1986. [Cited on pages 58 and 144.]
- Markus Langer, Cornelius J. König, Patrick Gebhard, and Elisabeth André. Dear Computer, Teach Me Manners: Testing Virtual Employment Interview Training. *International Journal of Selection and Assessment*, 24(4):312–323, 2016. [Cited on page 233.]
- Stephen R.H. Langton, Roger J. Watt, and Vicki Bruce. Do the eyes have it? Cues to the Direction of Social Attention. *Trends in Cognitive Sciences*, 4(2):50–59, February 2000. [Cited on pages 31 and 38.]
- Martha Larkin. *Using Scaffolded Instruction To Optimize Learning*. Eric Digests, ERIC Clearinghouse on Disabilities and Gifted Education Arlington VA, USA, 2002. [Cited on page 231.]
- Marc E. Latoschik. *Multimodale Interaktion in Virtueller Realität am Beispiel der Virtuellen Konstruktion*. PhD thesis, Bielefeld University, Germany, 2001. [Cited on page 95.]
- Marc E. Latoschik. Designing Transition Networks for Multimodal VR-Interactions Using a Markup Language. In *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces, ICMI 2002, Pittsburgh, PA, USA, October 14-16, 2002*, pages 411–416. IEEE Computer Society, Washington, DC, USA, 2002. [Cited on pages 95, 118, and 133.]
- Marc E. Latoschik. A User Interface Framework for Multimodal VR Interactions. In *Proceedings of the 7th International Conference on Multimodal Interfaces, Toronto, Italy, October 04 - 06, 2005*, pages 76–83. ACM New York, NY, USA, 2005. [Cited on pages 95, 99, 118, and 133.]
- Jina Lee and Stacy Marsella. Modeling Side Participants and Bystanders: The Importance of Being a Laugh Track. In Hannes Högni Vilhjálmsson, Stefan Kopp, Stacy Marsella, and Kristinn R. Thórisson, editors, *Proceedings of the 11th International Conference on Intelligent Virtual Agents, IVA 2011, Reykjavik, Iceland, September 15 - 17, 2011*, volume 6895 of *Lecture Notes in Computer Science*, pages 240–247. Springer-Verlag, Berlin, Heidelberg, 2011. [Cited on page 183.]

- Jina Lee, Stacy Marsella, David Traum, Jonathan Gratch, and Brent Lance. The Rickel Gaze Model: A Window on the Mind of a Virtual Human. In *Proceedings of the 7th International Conference on Intelligent Virtual Agents, IVA 2007, Paris, France, September 17-19, 2007*, volume 4722 of *Lecture Notes in Computer Science*, pages 296–303. Springer-Verlag Berlin Heidelberg, 2007. [Cited on pages 37, 77, and 99.]
- Ann Leffler, Dair L. Gillespie, and Joseph C. Conaty. The Effects of Status Differentiation on Nonverbal Behavior. *Social Psychology Quarterly*, 45(3):153–161, September 1982. [Cited on page 39.]
- Iolanda Leite, Carlos Martinho, and Ana Paiva. Social Robots for Long-Term Interaction: A Survey. *International Journal of Social Robotics*, 5(2):291–308, April 2013. [Cited on pages 6, 15, 63, and 68.]
- James C. Lester and Brian A. Stone. Increasing Believability in Animated Pedagogical Agents. In *Proceedings of the 1st International Conference on Autonomous Agents, AGENTS 1997, Marina del Rey, California, USA, February 05 - 08, 1997*, pages 16–21. ACM, New York, NY, USA, 1997. [Cited on page 81.]
- James C. Lester, Jennifer L. Voerman, Stuart G. Towns, and Charles B. Callaway. Cosmo: A Life-like Animated Pedagogical Agent with Deictic Believability. In *Proceedings of IJCAI97 Workshop on Animated Interface Agents, Nagoya, Japan, August 25, 1997*, 1997. [Cited on page 81.]
- Willem J. M. Levelt. Monitoring and Self-Repair in Speech. *Journal of Cognition*, 14(1):41–104, July 1983. [Cited on page 23.]
- Joan A. Levin and Johanna A. Moore. Dialogue Games: Metacommunication Structures for Natural Language Interaction. *Journal of Cognitive Science*, 1(4):395–420, 1977. [Cited on page 83.]
- Stephen C. Levinson and Francisco Torreira. Timing in Turn-Taking and its Implications for Processing Models of Language. *Frontiers in Psychology*, 6(731):1–17, June 2015. [Cited on page 39.]
- Steven C. Levinson. Putting Linguistics on a Proper Footing: Explorations in Goffman’s Participation Framework. In P. Drew and A. Wootton, editors, *Goffman: Exploring the Interaction Order*, pages 161–227. Oxford: Polity Press, 1988. [Cited on page 32.]
- Stephen C. Levinson. *Deixis*, pages 97–121. Blackwell Publishing Ltd, 2008. [Cited on page 162.]
- Han Z. Li, Young ok Yum, Robin Yates, Laura Aguilera, Ying Mao, and Yue Zheng. Interruption and Involvement in Discourse: Can Intercultural Interlocutors be Trained? *Journal of Intercultural Communication Research*, 34(4):233–254, December 2004. [Cited on page 43.]
- Han Z. Li. Cooperative and Intrusive Interruptions in Inter- and Intracultural Dyadic Discourse. *Journal of Language and Social Psychology*, 20(3):259–284, September 2001. [Cited on pages 41, 43, 60, and 175.]
- Per Lindström. First-Order Predicate Logic with Generalized Quantifiers. *Theoria*, 32:186–195, 1966. [Cited on page 129.]
- Lindsay Lipscomb, Janet Swanson, and Anne West. Scaffolding. In Michael Orey, editor, *Emerging Perspectives on Learning, Teaching, and Technology*. Department of Educational Psychology and Instructional Technology, University of Georgia, 2001. [Cited on page 231.]
- Changsong Liu, Rui Fang, and Joyce Y. Chai. Shared Gaze in Situated Referential Grounding: An Empirical Study. In Y. I. Nakano, C. Conati, and T. Bader, editors, *Eye Gaze in Intelligent User Interfaces: Gaze-based Analyses, Models and Applications*, pages 23–39. Springer-Verlag London, 2013. [Cited on pages 27, 29, and 66.]
- Markus Löckelt, Norbert Pflieger, and Norbert Reithinger. Multi-party Conversation for Mixed Reality. *International Journal of Virtual Reality*, 06(4):31–42, December 2007. [Cited on page 83.]
- Max M. Louwerse, Rick Dale, Ellen G. Bard, and Patrick Jeuniaux. Behavior Matching in Multimodal Communication Is Synchronized. *Journal of Cognitive Science*, 36(8):1404–1426, November–December 2012. [Cited on pages 18 and 182.]
- A. Bryan Loyall. *Believable Agents: Building Interactive Personalities*. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, 1997. [Cited on page 10.]
- David C. Luckham. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001. [Cited on page 118.]

- Lutz Lukas, Felix Schwägerl, and Marc Erich Latoschik. Unifikationsbasierte Sprach-Gesten Fusion für Multimodale VR/AR-Schnittstellen. In *Virtuelle und Erweiterte Realität, 7. Workshop of the GI Special Interest Group VR/AR*, pages 145–156. Shaker Verlag, 2010. [Cited on page 92.]
- Joanne Lumsden, Lynden K. Miles, Michael J. Richardson, Carlene A. Smith, and Neil Macrae. Who syncs? Social Motives and Interpersonal Coordination. *Journal of Experimental Social Psychology*, 48(3):746–751, May 2012. [Cited on pages 15, 20, and 21.]
- Robert MacDonald. Disconnected youth? Social Exclusion, the ‘Underclass’ and Economic Marginality. *Journal of Social Work and Society*, 6(2):236–248, 2008. [Cited on page 233.]
- C. Neil Macrae, Oonagh K. Duffy, Lynden K. Miles, and Julie Lawrence. A Case of Hand Waving: Action Synchrony and Person Perception. *Journal of Cognition*, 109(1):152–156, October 2008. [Cited on page 21.]
- François Mairesse and Marilyn A. Walker. Controlling User Perceptions of Linguistic Style: Trainable Generation of Personality Traits. *Journal of Computational Linguistics, MIT Press Cambridge, MA, USA*, 37(3):455–488, September 2011. [Cited on page 71.]
- Stacy C. Marsella, W. Lewis Johnson, and Catherine LaBore. Interactive Pedagogical Drama. In *Proceedings of the 4th International Conference on Autonomous Agents, AGENTS 2000, Barcelona, Spain, June 3 - 7, 2000*, pages 301–308. ACM New York, NY, USA, 2000. [Cited on pages 87 and 231.]
- Stacy C. Marsella, W. Lewis Johnson, and Catherine LaBore. Interactive Pedagogical Drama for Health Interventions. In *Proceedings of the 11th International Conference on Artificial Intelligence in Education, AIED 2003, Sydney, Australia, July 20-24, 2003*, pages 341–348. IOS Press, 2003. [Cited on page 231.]
- Kerry L. Marsh, Michael J. Richardson, and Richard C. Schmidt. Social Connection Through Joint Action and Interpersonal Coordination. *Topics in Cognitive Science*, 1(2):320–339, April 2009. [Cited on pages 17, 21, and 28.]
- Malia F. Mason, Elizabeth P. Tatkov, and C. Neil Macrae. The Look of Love: Gaze Shifts and Person Perception. *Journal of Psychological Science*, 16(3):236–239, March 2005. [Cited on page 38.]
- Michael Mateas and Andrew Stern. Towards Integrating Plot and Character for Interactive Drama. In Kerstin Dautenhahn, Alan Bond, Lola Cañamero, and Bruce Edmonds, editors, *Socially Intelligent Agents: Creating Relationships with Computers and Robots*, volume 3 of *Multiagent Systems, Artificial Societies, and Simulated Organizations*, pages 221–228. Springer US, 2002. [Cited on page 233.]
- Michael Mateas and Andrew Stern. Facade: An Experiment in Building a Fully-Realized Interactive Drama. In *Proceedings of the Game Developer’s Conference: Game Design Track, San Jose, California, March 2003.*, 2003. [Cited on page 84.]
- Michael Mateas and Andrew Stern. Facade: An Experiment in Building a Fully-Realized Interactive Drama. In *Game Developer’s Conference: Game Design Track, San Jose, California, USA, March 2003*, 2003. [Cited on page 231.]
- Josh McCoy, Mike Treanor, Ben Samuel, Brandon Tearse, Michael Mateas, and Noah Wardrip-Fruin. Authoring Game-Based Interactive Narrative using Social Games and Comme il Faut. In *Proceedings of the 4th International Conference and Festival of the Electronic Literature Organization: Archive and Innovate, ELO 2010, Providence, Rhode Island, USA, June, 2010*, 2010. [Cited on page 233.]
- Gregory J. McHugo, John T. Lanzetta, Denis G. Sullivan, Roger D. Masters, and Basil G. Englis. Emotional Reactions to a Political Leader’s Expressive Displays. *Journal of Personality and Social Psychology*, 49(6):1513–1529, December 1985. [Cited on page 19.]
- D. McNeill. *Hand and Mind: What Gestures Reveal about Thought*. University of Chicago Press, 1992. [Cited on page 232.]
- Michael F. McTear. Modeling Spoken Dialogues with State Transition Diagrams: Experiences wit the CSLU Toolkit. In *Proceedings of the 5th International Conference on Spoken Language Processing, ICSLP 1998, Sydney, Australia, November 30 - December 4, 1998*, pages 1223–1226, 1998. [Cited on pages 85 and 133.]
- Michael F. McTear. Using the CSLU Toolkit for Practicals in Spoken Dialogue Technology. In *Proceedings of the ESCA/SOCRATES Workshop on Method and Tool Innovations for Speech Science Education (MATISSE), University College, London, UK, April 16-17, 1999*, pages 113–116, 1999. [Cited on page 85.]

- George H. Mealy. A Method to Synthesizing Sequential Circuits. *Bell System Technical Journal*, pages 1045–1079, 1995. [Cited on page 88.]
- Gregor U. Mehlmann and Elisabeth André. Modeling Multimodal Integration with Event Logic Charts. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction, ICMI 2012, Santa Monica, California, USA, October 22-26, 2012*, ACM International Conference Proceedings, pages 125–132. ACM, New York, NY, USA, 2012. [Cited on pages 52, 64, 65, 118, 119, and 200.]
- Gregor U. Mehlmann, Markus Häring, René Bühling, Michael Wißner, and Elisabeth André. Multiple Agent Roles in an Adaptive Virtual Classroom Environment. In Jan M. Allbeck, Norman I. Badler, Timothy W. Bickmore, Catherine Pelachaud, and Alla Safonova, editors, *Proceedings of the 10th International Conference on Intelligent Virtual Agents, IVA 2010, Philadelphia, Pennsylvania, USA, September 20-22, 2010*, volume 6356 of *Lecture Notes in Computer Science*, pages 250–256. Springer-Verlag, Berlin, Heidelberg, 2010. [Cited on page 230.]
- Gregor U. Mehlmann, Birgit Endrass, and Elisabeth André. Modeling Parallel State Charts for Multithreaded Multimodal Dialogues. In *Proceedings of the 13th International Conference on Multimodal Interaction, ICMI 2011, Alicante, Spain, November 14-18, 2011*, ACM International Conference Proceedings, pages 385–392. ACM, New York, NY, USA, 2011. [Cited on page 200.]
- Gregor U. Mehlmann, Patrick Gebhard, Birgit Endrass, and Elisabeth André. SceneMaker: Visual Authoring of Dialogue Processes. In *Proceedings of the 7th Workshop on Knowledge and Reasoning in Practical Dialogue Systems, KRPS 2011, held at the 22th International Joint Conference on Artificial Intelligence, IJCAI 2011, Barcelona, Spain, July 16-22, 2011*, pages 24–36, 2011. [Cited on pages 190 and 200.]
- Gregor U. Mehlmann, Kathrin Janowski, Tobias Baur, Markus Häring, Elisabeth André, and Patrick Gebhard. Exploring a Model of Gaze for Grounding in HRI. In *Proceedings of the 16th ACM International Conference on Multimodal Interaction, ICMI 2014, Istanbul, Turkey, November 12-16, 2014*, ACM International Conference Proceedings, pages 247–254. ACM, New York, NY, USA, 2014. [Cited on pages 28, 30, 65, 66, 119, 151, 154, 159, 172, 190, 200, 235, and 237.]
- Gregor U. Mehlmann, Kathrin Janowski, Tobias Baur, Elisabeth André, Markus Häring, and Patrick Gebhard. Modeling Gaze Mechanisms for Grounding in HRI. In *Proceedings of the 21th European Conference on Artificial Intelligence, ECAI 2014, Prague, Czech Republic, August 18-22, 2014*, *Frontiers in Artificial Intelligence and Applications*, pages 1069–1070. IOS Press Ebooks, Amsterdam, The Netherlands, 2014. [Cited on pages 15, 28, 30, 42, 151, 154, 172, 200, 235, and 237.]
- Gregor Mehlmann, Kathrin Janowski, and Elisabeth André. Modeling Grounding for Interactive Social Companions. *Special Issue of the German Journal of Artificial Intelligence: Social Companion Technologies*, Springer-Verlag Berlin Heidelberg, 30(1):45–52, September 2016. [Cited on pages 10, 30, 65, 118, 119, 151, 153, 154, 200, 235, and 237.]
- Gregor U. Mehlmann. *Scenemaker 3: An Interpreter for Parallel Processes Modeling Behavior of Interactive Virtual Characters*. Master's Thesis in Computer Science, German Research Center for Artificial Intelligence, Saarland University, Faculty of Natural Sciences and Technology I, Department of Computer Science, June 2009. [Cited on pages 85, 148, 199, 200, and 201.]
- Albert Mehrabian. *Non-Verbal Communication*. Aldine-Atherton, Chicago, Illinois, 1972. [Cited on page 56.]
- Chris Mellish and Gerald Gazdar. *Natural Language Processing in PROLOG: An Introduction to Computational Linguistics*. Addison-Wesley, 1989. [Cited on pages 120 and 121.]
- Leo Meltzer, William N. Morris, and Donald P. Hayes. Interruption Outcomes and Vocal Amplitude: Explorations in Social Psychophysics. *Journal of Personality and Social Psychology*, 18(3):392–402, June 1971. [Cited on pages 39 and 40.]
- Andrew N. Meltzoff and Rechele Brooks. "Like me" As a Building Block for Understanding other Minds: Bodily Acts, Attention, and Intention. In Bertram Malle, L. J. Moses, and Dare Baldwin, editors, *Intentions and Intentionality: Foundations of Social Cognition.*, pages 171–191. MIT Press, Cambridge, MA, USA, 2001. [Cited on pages 5, 33, and 54.]
- Andrew N. Meltzoff and M. Keith Moore. Newborn Infants Imitate Adult Facial Gestures. *Child Development*, 54(3):702–709, June 1983. [Cited on pages 18 and 19.]
- Antje S. Meyer, Astrid M. Sleiderink, and Willem J. M. Levelt. Viewing and Naming Objects: Eye Movements During Noun Phrase Production. *Journal of Cognition*, 66(2):B25–B33, May 1998. [Cited on pages 5, 34, 62, 65, and 67.]

- Jan Miksatko and Michael Kipp. Hybrid Control for Embodied Agents Applications. In *Proceedings of the 32nd Annual German Conference on Advances in Artificial Intelligence, KI 2009, Paderborn, Germany, September 15-18, 2009*, pages 524–531. Springer-Verlag Berlin, Heidelberg, 2009. [Cited on page 88.]
- Jan Miksatko, Kerstin H. Kipp, and Michael Kipp. The Persona Zero-Effect: Evaluating Virtual Character Benefits on a Learning Task with Repeated Interactions. In Jan Allbeck, Norman Badler, Timothy Bickmore, Catherine Pelachaud, and Alla Safonova, editors, *Proceedings of the 10th International Conference on Intelligent Virtual Agents, IVA 2010, Philadelphia, PA, USA, September 20-22, 2010*, pages 475–481. Springer Berlin Heidelberg, 2010. [Cited on page 88.]
- Lynden K. Miles, Louise K. Nind, and C. Neil Macrae. The Rhythm of Rapport: Interpersonal Synchrony and Social Perception. *Journal of Experimental Social Psychology*, 45(3):585–589, 2009. [Cited on pages 21 and 29.]
- Lynden K. Miles, Jordan L. Griffiths, Michael J. Richardson, and C. Neil Macrae. Too Late to Coordinate: Contextual Influences on Behavioral Synchrony. *European Journal of Social Psychology*, 40(1):52–60, February 2010. [Cited on page 21.]
- Lynden K. Miles, Louise K. Nind, Zoe Henderson, and C. Neil Macrae. Moving Memories: Behavioral Synchrony and Memory for Self and Others. *Journal of Experimental Social Psychology*, 46(2):457–460, 2010. [Cited on page 21.]
- Lynden K. Miles, Joanne Lumsden, Michael J. Richardson, and C. Neil Macrae. Do Birds of a Feather Move Together? Group Membership and Behavioral Synchrony. *Journal of Experimental Brain Research*, 211(3-4):495–503, June 2011. [Cited on page 21.]
- Marvin Minski. A Framework for Representing Knowledge. In Patrick Henry Winston, editor, *Default Reasoning and Lexical Organization, The Psychology of Computer Vision*, pages 211–277. McGraw-Hill, New York, NY, USA, 1975. [Cited on pages 90 and 119.]
- Mehryar Mohri, Fernando Pereira, and Michael Riley. Weighted Finite-State Transducers in Speech Recognition. *Computer Speech and Language*, 16(1):69–88, 2002. [Cited on page 94.]
- Akito Monden, Kenichi Matsumoto, and Masatake Yamato. Evaluation of Gaze-Added Target Selection Methods Suitable for General GUIs. *International Journal of Computer Applications in Technology*, 24(1):17–24, June 2005. [Cited on pages 66 and 159.]
- Richard Montague. The Proper Treatment of Quantification in Ordinary English. In Jack Kulas, James H. Fetzer, and Terry L. Rankin, editors, *Philosophy, Language, and Artificial Intelligence: Resources for Processing Natural Language*, volume 2 of *Studies in Cognitive Systems*, pages 141–162. Springer Netherlands, Dordrecht, 1988. [Cited on pages 108 and 130.]
- Andrzej Mostowski. On a Generalization of Quantifiers. *Fundamenta Mathematicae*, 44(1):12–36, 1957. [Cited on page 129.]
- Samer Al Moubayed, Jonas Beskow, Gabriel Skantze, and Björn Granström. Furhat: A Back-Projected Human-Like Robot Head for Multiparty Human-Machine Interaction. In Anna Esposito, Antonietta M. Esposito, Alessandro Vinciarelli, Rüdiger Hoffmann, and Vincent C. Müller, editors, *Cognitive Behavioural Systems: COST 2102 International Training School, Dresden, Germany, February 21-26, 2011, Revised Selected Papers*, pages 114–130. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. [Cited on page 86.]
- Samer Al Moubayed, Gabriel Skantze, and Jonas Beskow. The Furhat Back-Projected Humanoid Head - Lip Reading, Gaze and Multiparty Interaction. *International Journal on Humanoid Robotics*, 10(1):25, March 2013. [Cited on pages 74, 76, and 86.]
- Peter Mundy and Lisa Newell. Attention, Joint Attention, and Social Cognition. *Current Directions in Psychological Science*, 16(5):269–274, October 2007. [Cited on pages 27, 29, 33, 54, 56, 59, and 75.]
- Tadao Murata. Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, 77(4):541 – 580, April 1989. [Cited on pages 87 and 95.]
- Kumiko Murata. Intrusive or co-operative? a cross-cultural study of interruption. *Journal of Pragmatics*, 21(4):385–400, 1994. [Cited on pages 41 and 173.]
- Kevin Patrick Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California, Berkeley, 2002. [Cited on page 153.]

- Stephen O. Murray. Toward a Model of Members' Methods for Recognizing Interruptions. *Journal of Language in Society*, 14(1):31–40, March 1985. [Cited on pages 40 and 41.]
- Bilge Mutlu, Jodi Forlizzi, and Jessica Hodgins. A Storytelling Robot: Modeling and Evaluation of Human-Like Gaze Behavior. In *In Proceedings of the 6th IEEE-RAS International Conference on Humanoid Robots, Humanoids 2006, Genova, Italy, December 4–6, 2006*, pages 518–523, 2006. [Cited on page 75.]
- Bilge Mutlu, Toshiyuki Shiwa, Takayuki Kanda, Hiroshi Ishiguro, and Norihiro Hagita. Footing In Human-Robot Conversations: How Robots Might Shape Participant Roles Using Gaze Cues. In *Proceedings of the 4th ACM/IEEE International Conference on Human-Robot Interaction, HRI 2009, La Jolla, CA, USA, March 9–13, 2009*, 2009. [Cited on pages 32, 74, and 77.]
- Bilge Mutlu, Takayuki Kanda, Jodi Forlizzi, Jessica Hodgins, and Hiroshi Ishiguro. Conversational Gaze Mechanisms for Humanlike Robots. *ACM Transactions on Interactive Intelligent Systems, TIIS*, 1(2), January 2012. [Cited on pages 15, 74, 77, 99, and 193.]
- Bilge Mutlu, Allison Terrell, and Chien-Ming Huang. Coordination Mechanisms in Human-Robot Collaboration. In *Proceedings of the Workshop on Collaborative Manipulation, 8th ACM/IEEE International Conference on Human-Robot Interaction, HRI 2013, Tokyo, Japan, March 3–6, 2013*, 2013. [Cited on pages 70, 74, 75, and 99.]
- Lee Naish. Higher-Order Logic Programming in Prolog. Technical Report 96/2, Department of Computer Science, University of Melbourne, Melbourne, Australia, February 1996. [Cited on pages 67 and 108.]
- Yukiko I. Nakano and Ryo Ishii. Estimating User's Engagement from Eye-Gaze Behaviors in Human-Agent Conversations. In *Proceedings of the 15th International Conference on Intelligent User Interfaces, IUI 2010, Hong Kong, China, February 7–10, 2010*, pages 139–148. ACM New York, NY, USA, 2010. [Cited on page 34.]
- Yukiko I. Nakano, Gabe Reinstein, Tom Stocky, and Justine Cassell. Towards a Model of Face-to-Face Grounding. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, ACL 2003, Saporro, Japan July 7–12, 2003*, pages 553–561. Association for Computational Linguistics Stroudsburg, PA, USA, 2003. [Cited on pages 28, 74, and 75.]
- Michael Natale, Elliot Entin, and Joseph Jaffe. Vocal Interruptions in Dyadic Communication as a Function of Speech and Social Anxiety. *Journal of Personality and Social Psychology*, 37(6):865–878, June 1979. [Cited on pages 40, 187, and 195.]
- Dana Nau, Yue Cao, Amnon Lotem, and Hector Munoz-Avila. SHOP: Aimple Hierarchical Ordered Planner. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence, IJCAI 1999, Stockholm, Sweden, July 31 - August 6, 1999*, pages 968–973. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1999. [Cited on page 83.]
- Dana Nau, Okhtay Ilghami, Ugur Kuter, J. William Murdock, Dan Wu, and Fusun Yaman. SHOP2: An HTN Planning System. *Journal of Artificial Intelligence Research*, 20:379–404, 2003. [Cited on page 83.]
- David Navarre, Philippe Palanque, Rémi Bastide, Amélie Schyn, Marco Winckler, Luciana P. Nedel, and Carla M.D.S. Freitas. A Formal Description of Multimodal Interaction Techniques for Immersive Virtual Reality Applications. In *IFIP TC13 International Conference on Human-Computer Interaction, INTERACT 2005, Rome, Italy, September 12–16, 2005*, pages 170–183. Springer Berlin-Heidelberg, 2005. [Cited on pages 95 and 133.]
- Alassane Ndiaye, Patrick Gebhard, Michael Kipp, Martin Klesen, Michael Schneider, and Wolfgang Wahlster. Ambient Intelligence in Edutainment: Tangible Interaction with Life-Like Exhibit Guides. In *Proceedings of the 1st International Conference on Intelligent Technologies for Interactive Entertainment, INTETAIN 2005, Madonna di Campiglio, Italy, November 30 - December 2, 2005*, volume 3814 of *Lecture Notes in Computer Science*, pages 104–113. Springer-Verlag, Berlin, Heidelberg, 2005. [Cited on pages 85 and 199.]
- Jeannette G. Neal, Zuzana Krifka Dobes, Keith E. Bettinger, and Jong S. Byoun. Multi-Modal References in Human-Computer Dialogue. In Howard E. Shrobe, Tom M. Mitchell, and Reid G. Smith, editors, *Proceedings of the 7th National Conference on Artificial Intelligence, St. Paul, MN, USA, August 21–26, 1988*, pages 819–825, 1988. [Cited on page 89.]
- Jeannette G. Neal, C. Y. Thielman, Zuzana Dobes, S. M. Haller, and Stuart C. Shapiro. Natural Language with Integrated Deictic and Graphic Gestures. In *Proceedings of the Workshop on Speech and Natural Language, HLT 1989*, pages 410–423. Association for Computational Linguistics, Stroudsburg, PA, USA, 1989. [Cited on page 89.]

- Sik Nung Ng, Mark Brooke, and Michael Dunne. Interruption and Influence in Discussion Groups. *Journal of Language and Social Psychology*, 14(4):369–381, December 1995. [Cited on page 41.]
- Keith A. Nichols and Brian G. Champness. Eye Gaze and the GSR. *Journal of Experimental Social Psychology*, 7(6):623–626, November 1971. [Cited on pages 34 and 35.]
- Gerhard Nielsen. *Studies in Self Confrontation*. Munksgaard, Copenhagen, Denmark, 1962. [Cited on pages 5, 17, 21, 35, 55, 193, and 235.]
- Laurence Nigay and Joëlle Coutaz. A Design Space for Multimodal Systems: Concurrent Processing and Data Fusion. In *Proceedings of the INTERACT 1993 and CHI 1993 Conference on Human Factors in Computing Systems, Amsterdam, The Netherlands*, pages 172–178. ACM, New York, NY, 1993. [Cited on page 90.]
- Laurence Nigay and Joëlle Coutaz. A Generic Platform for Addressing the Multimodal Challenge. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 1995, Denver, Colorado, USA, May 07–11, 1995*, pages 98–105, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1995. [Cited on page 90.]
- Tsukasa Noma and Norman Badler. A Virtual Human Presenter. In *Proceedings of IJCAI97 Workshop on Animated Interface Agents, Nagoya, Japan, August 25, 1997*, 1997. [Cited on page 86.]
- Tsukasa Noma, Liwei Zhao, and Norman Badler. Design of a Virtual Human Presenter. *IEEE Computer Graphics and Applications Journal*, IEEE Computer Society Press Los Alamitos, CA, USA, 20(4):79–85, July 2000. [Cited on page 86.]
- Bahador Nooraei, Charles Rich, and Candace L. Sidner. A Real-Time Architecture for Embodied Conversational Agents: Beyond Turn-Taking. In *Proceedings of the 7th International Conference on Advances in Computer-Human Interactions, ACHI 2014, Barcelona, Spain, March 23–27, 2014*, pages 381–388, 2014. [Cited on pages 87 and 190.]
- Florian Nothdurft, Gregor Behnke, Pascal Bercher, Susanne Biundo, and Wolfgang Minker. The Interplay of User-Centered Dialog Systems and AI Planning. In *Proceedings of the 16th Annual SIGdial Meeting on Discourse and Dialogue, SIGDIAL 2015, Prague, Czech Republic, September 2–4, 2015*, pages 344–353, 2015. [Cited on page 83.]
- Mary Octigan and Sharon Niederman. Male Dominance in Conversations. *Frontiers*, 4(1):50–54, 1979. [Cited on page 44.]
- Catharine Oertel, Marcin Włodarczak, Jens Edlund, Petra Wagner, and Joakim Gustafson. Gaze Patterns in Turn-Taking. In *Proceedings of the 13th Annual Conference of the International Speech Communication Association 2012, INTERSPEECH 2012, Portland, Oregon, USA, September 9–13, 2012*, pages 2243–2246, 2012. [Cited on pages 36 and 181.]
- Yuko Okumura, Yasuhiro Kanakogi, Takayuki Kanda, Hiroshi Ishiguro, and Shoji Itakura. Infants Understand the Referential Nature of Human Gaze but not Robot Gaze. *Journal of Experimental Child Psychology*, 116(1):86–95, September 2013. [Cited on page 77.]
- Margarethe Olbertz-Siitonen. *Unterbrechen in zielgerichteten Gesprächen*. PhD thesis, Faculty of Humanities of the University of Tampere, 2009. [Cited on pages 15 and 39.]
- Bengt Orestrom, editor. *Turn-Taking in English Conversation*. CWK Gleerup Lund, 1983. [Cited on pages 42 and 43.]
- James P. Otteson and Carol R. Otteson. Effect of Teacher’s Gaze on Children’s Story Recall. *Perceptual and Motor Skills*, 50(1):35–42, February 1980. [Cited on page 35.]
- John K. Ousterhout. Scripting: Higher Level Programming for the 21st Century. *Computer*, IEEE Computer Society, 31(3):23–30, March 1998. [Cited on page 108.]
- Sharon Oviatt and Philip Cohen. Multimodal Interfaces That Process What Comes Naturally. *Communications of the ACM*, 43:45–53, 2000. [Cited on page 34.]
- Sharon Oviatt and Robert VanGent. Error Resolution During Multimodal Human-Computer Interaction. In *Proceedings of the Fourth International Conference on Spoken Language, ICSLP 1996, Philadelphia, PA, USA, October 3–6, 1996*, pages 204–207. ACM, New York, NY, USA, 1996. [Cited on page 34.]
- Sharon Oviatt, Antonella DeAngeli, and Karen Kuhn. Integration and Synchronization of Input Modes During Multimodal Human-computer Interaction. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, CHI 1997, Atlanta, Georgia, USA, March 22–27, 1997*, pages 415–422. ACM, New York, NY, USA, 1997. [Cited on pages 34, 52, and 55.]

- Sharon Oviatt, Phil Cohen, Lizhong Wu, Lisbeth Duncan, Bernhard Suhm, Josh Bers, Thomas Holzman, Terry Winograd, James Landay, Jim Larson, and David Ferro. Designing the User Interface for Multimodal Speech and Pen-Based Gesture Applications: State-of-the-Art Systems and Future Research Directions. *Journal on Human-Computer Interaction*, 15(4):263–322, December 2000. [Cited on pages 34, 89, and 119.]
- Sharon Oviatt, Kevin Hang, Jianlong Zhou, and Fang Chen. Spoken Interruptions Signal Productive Problem Solving and Domain Expertise in Mathematics. In Zhengyou Zhang, Phil Cohen, Dan Bohus, Radu Horaud, and Helen Meng, editors, *Proceedings of the 17th ACM on International Conference on Multimodal Interaction, ICMI 2015, Seattle, Washington, USA, November 9-13, 2015*, pages 311–318. ACM New York, NY, USA, 2015. [Cited on pages 5, 43, 44, 173, and 235.]
- Sharon Oviatt, Björn Schuller, Philip R. Cohen, Daniel Sonntag, Gerasimos Potamianos, and Antonio Krüger, editors. *The Handbook of Multimodal-Multisensor Interfaces: Foundations, User Modeling, and Common Modality Combinations - Volume 1*. Association for Computing Machinery and Morgan and Claypool, New York, NY, USA, 2017. [Cited on page 89.]
- Sharon Oviatt. Multimodal Interfaces for Dynamic Interactive Maps. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 1996, Vancouver, British Columbia, Canada, April 13-18, 1996*, pages 95–102. ACM, New York, NY, USA, 1996. [Cited on page 34.]
- Sharon Oviatt. Multimodal Interactive Maps: Designing for Human Performance. *Journal on Human Computer Interaction, Special Issue on Multimodal Interfaces*, 12(1):93–129, March 1997. [Cited on page 34.]
- Sharon Oviatt. Mutual Disambiguation of Recognition Errors in a Multimodal Architecture. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 1999, Pittsburgh, Pennsylvania, USA*, pages 576–583. ACM, New York, NY, USA, 1999. [Cited on pages 34, 56, 65, and 92.]
- Sharon Oviatt. Breaking the Robustness Barrier: Recent Progress on the Design of Robust Multimodal Systems. *Journal on Advances in Computers*, 56:305–341, 2002. [Cited on page 34.]
- Sharon Oviatt. Advances in Robust Multimodal Interface Design. *IEEE Computer Graphics and Applications*, 23(5):62–68, September 2003. [Cited on pages 26, 34, and 235.]
- Sharon Oviatt. Multimodal Interfaces. In Andrew Sears and Julie A. Jacko, editors, *The Human-Computer Interaction Handbook: Fundamentals, Evolving techniques and Emerging Applications*, pages 405–430. Lawrence Erlbaum, Mahwah, NJ, USA, 2012. [Cited on pages 26, 27, 34, 56, 62, 65, 68, and 119.]
- Oleg Špakov. Comparison of Gaze-to-Objects Mapping Algorithms. In *Proceedings of the 1st Conference on Novel Gaze-Controlled Applications, NGCA 2011, Karlskrona, Sweden, May 26 - 27, 2011*. ACM New York, NY, USA, 2011. [Cited on pages 66 and 159.]
- Maria-Paola Paladino, Mara Mazzurega, Francesco Pavani, and Thomas W. Schubert. Synchronous Multisensory Stimulation Blurs Self-Other Boundaries. *Journal of Psychological Science*, 21(9):1202–1207, September 2010. [Cited on page 21.]
- Miles L. Patterson. An Arousal Model of Interpersonal Intimacy. *Journal of Psychological Review*, 83(3):235–245, May 1976. [Cited on pages 34, 35, and 38.]
- Catherine Pelachaud. Multimodal Expressive Embodied Conversational Agents. In *Proceedings of the 13th Annual ACM International Conference on Multimedia, MULTIMEDIA 2005, Hilton, Singapore, November 06 - 11, 2005*, pages 683–689. ACM, New York, NY, USA, 2005. [Cited on pages 6, 15, 96, 99, 110, and 232.]
- Francis Jeffrey Pelletier. Did Frege Believe Frege’s Principle? *Journal of Logic, Language and Information*, 10(1):87–114, March 2001. [Cited on page 141.]
- Kevin A. Pelphrey, Ronald J. Viola, and Gregory McCarthy. When Strangers Pass: Processing of Mutual and Averted Social Gaze in the Superior Temporal Sulcus. *Journal of Psychological Science*, 15(9):598–603, September 2004. [Cited on page 31.]
- Fernando C. N. Pereira and Stuart M. Shieber. *PROLOG and Natural Language Analysis*. CSLI Publications, Stanford, CA, USA, 1987. [Cited on page 108.]
- Fernando Pereira. Logic for Natural Language Analysis. Technical report, Artificial intelligence Center, Computer Science and Technology Division, University of Edinburgh, 1983. [Cited on page 130.]

- Fernando C. N. Pereira. Logic for Natural Language Analysis. Ph. D. thesis, University of Edinburgh, Edinburgh, Scotland, Reprinted as Technical Note 275, January 1983, Artificial Intelligence Center, SRI International, Menlo park, California, 1983. [Cited on page 108.]
- Fernando Pereira. Review of “The logic of typed feature structures” by Bob Carpenter, Cambridge University Press 1992. *Journal of Computational Linguistics*, MIT Press Cambridge, MA, USA, 19(3):544–552, September 1993. [Cited on pages 63 and 118.]
- Ken Perlin and Athomas Goldberg. Improv: A System for Scripting Interactive Actors in Virtual Worlds. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1996*, New Orleans, Louisiana, USA, 1996, pages 205–216. ACM, New York, NY, USA, 1996. [Cited on page 81.]
- David I. Perrett and Nathan J. Emery. Understanding the Intentions of Others from Visual Signals. *Current Psychology of Cognition*, 13(5):683–694, October 1994. [Cited on page 34.]
- Natalie K. Person, Arthur C. Graesser, Derek Harter, and Eric Mathews. Dialog Move Generation and Conversation Management in AutoTutor. In *Building Dialog Systems for Tutorial Applications—Papers from the AAAI Fall Symposium*, pages 45–51, 2000. [Cited on page 85.]
- Stanley Peters and Dag Westerståhl. *Quantifiers in Language and Logic*. Claredon Press, Oxford, New York, 2006. [Cited on page 130.]
- Christopher Peters, Stylianos Asteriadis, and Kostas Karpouzis. Investigating Shared Attention with a Virtual Agent using a Gaze-Based Interface. *Journal on Multimodal User Interfaces*, 3(1-2):119–130, March 2010. [Cited on page 74.]
- Nadine Pfeiffer-Lessmann, Thies Pfeiffer, and Ipke Wachsmuth. An Operational Model of Joint Attention -Timing of Gaze Patterns in Interactions between Humans and a Virtual Human. In Naomi Miyake, David Peebles, and Richard P. Cooper, editors, *Proceedings of the 34th Annual Conference of the Cognitive Science Society, Sapporo, Japan, August 1-4, 2012*, volume 34, pages 851–856. Cognitive Science Society, Austin, TX, USA, 2012. [Cited on pages 31, 74, 75, and 99.]
- Norbert Pflieger. Context-Based Multimodal Fusion. In *Proceedings of the 6th International Conference on Multimodal Interfaces, ICMI 2004, State College, PA, USA, October 13 - 15, 2004*, ICMI '04, pages 265–272. ACM, New York, NY, USA, 2004. [Cited on pages 94, 99, and 119.]
- Fiona G. Phelps, Gwyneth Doherty-Sneddon, and Hannah Warnock. Helping Children Think: Gaze Aversion and Teaching. *British Journal of Developmental Psychology*, 24(3):577–588, September 2006. [Cited on page 37.]
- Rosalind W. Picard. *Affective Computing*. MIT Press, Cambridge, Massachusetts, USA, 1997. [Cited on pages 56 and 63.]
- Isabella Poggi, Catherine Pelachaud, and Fiorella De Rosis. Eye Communication in a Conversational 3D Synthetic Agent. *Journal of AI Communications*, 3(13):169 – 181, 2000. [Cited on page 96.]
- Carl Pollard and Ivan A. Sag. Information-based Syntax and Semantics. Volume 1: Fundamentals. In *CSLI Lecture Notes 13*. CSLI Publications, Stanford, CA, USA, 1987. [Cited on pages 119 and 121.]
- Carl Pollard and Ivan A. Sag. Head-Driven Phrase Structure Grammar. In *Studies in Contemporary Linguistics*. University of Chicago Press, Chicago, IL, USA, 1994. [Cited on page 119.]
- Pilar Manchón Portillo, Guillermo Pérez García, and Gabriel Amores Carredano. Multimodal Fusion: A New Hybrid Strategy for Dialogue Systems. In *Proceedings of the 8th International Conference on Multimodal Interfaces, ICMI 2006, Banff, Alberta, Canada, November 02 - 04, 2006*, pages 357–363. ACM, New York, NY, USA, 2006. [Cited on pages 64 and 119.]
- Michael I. Posner. Orienting of Attention. *Quarterly Journal of Experimental Psychology*, 32(1):3–25, February 1980. [Cited on page 33.]
- Richard J. D. Power and Maria F. Dal Martello. Some Criticisms of Sacks, Schegloff, and Jefferson on turn taking. *Semiotica*, 58(1-2):29–40, 1986. [Cited on page 42.]
- Zahar Prasov and Joyce Y. Chai. What’s in a Gaze? The Role of Eye-Gaze in Reference Resolution in Multimodal Conversational Interfaces. In *Proceedings of the 13th International Conference on Intelligent User Interfaces, UII 2008, Maspalomas, Gran Canaria, Spain, September 8-16, 2008*, pages 20–29, 2008. [Cited on pages 66 and 76.]

- Zahar Prasov and Joyce Y. Chai. Fusing Eye Gaze with Speech Recognition Hypotheses to Resolve Exophoric References in Situated Dialogue. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, MIT Stata Center, Cambridge, Massachusetts, USA, October 9-11*, pages 471–481. Association for Computational Linguistics, Stroudsburg, PA, USA, 2010. [Cited on pages 66 and 77.]
- Helmut Prendinger and Mitsuru Ishizuka. SCREAM: Scripting Emotion-Based Agent Minds. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2002, Bologna, Italy, July 15 - 19, 2002*, pages 350–351. ACM, New York, NY, USA, 2002. [Cited on page 81.]
- Helmut Prendinger, Sylvain Descamps, and Mitsuru Ishizuka. Scripting Affective Communication with Life-Like Characters in Web-Based Interaction Systems. *Applied Artificial Intelligence*, 16(7-8):519–553, 2002. [Cited on page 81.]
- Helmut Prendinger, Santi Saeyor, and Mitsuru Ishizuka. MPML and SCREAM: Scripting the Bodies and Minds of Life-Like Characters. *Cognitive Technologies: Life-Like Characters*, pages 213–242, September 2004. [Cited on page 81.]
- Matthew Purver, Jonathan Ginzburg, and Patrick Healey. On the Means for Clarification in Dialogue. In Jan van Kuppevelt and Ronnie W. Smith, editors, *Current and New Directions in Discourse and Dialogue*, pages 235–255. Springer Netherlands, 2003. [Cited on page 70.]
- Shaolin Qu and Joyce Y. Chai. Incorporating Temporal and Semantic Information with Eye Gaze for Automatic Word Acquisition in Multimodal Conversational Systems. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Waikiki, Honolulu, Hawaii, October 25 - 27, 2008*, pages 244–253. Association for Computational Linguistics, Stroudsburg, PA, USA, 2008. [Cited on page 66.]
- Veronica C. Ramenzoni, Tehran J. Davis, Michael A. Riley, Kevin Shockley, and Aimee A. Baker. Joint Action in a Cooperative Precision Task: Nested Processes of Intrapersonal and Interpersonal Coordination. *Experimental Brain Research*, 211(13):447–457, April 2011. [Cited on page 17.]
- Josephine M. Randel, Barbara A. Morris, C. Douglas Wetzel, and Betty V. Whitehill. The Effectiveness of Games for Educational Purposes: A Review of Recent Research. *Journal of Simulation Gaming*, 23(3):261–276, September 1992. [Cited on page 231.]
- Antoine Raux and Maxine Eskenazi. A Multi-Layer Architecture for Semi-Synchronous Event-Driven Dialogue Management. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition Understanding, ASRU 2007, Kyoto, Japan, December 9-13*, pages 514–519, 2007. [Cited on page 84.]
- Antoine Raux and Maxine Eskénazi. A Finite-State Turn-Taking Model for Spoken Dialog Systems. In *Proceedings of the Conference on Human Language Technologies of the North American Chapter of the Association of Computational Linguistics, Boulder, Colorado, USA, May 31 - June 5, 2009*, pages 629–637. The Association for Computational Linguistics, 2009. [Cited on page 133.]
- Peter Rechenberg and Gustav Pomberger. *Informatik Handbuch*. Carl Hanser Verlag, Wien, 1997. [Cited on page 217.]
- Matthias Rehm, Nikolas Bee, Birgit Endrass, Michael Wißner, and E. André. Too close for comfort? Adapting to the User’s Cultural Background. In *Proceedings of the International Workshop on Human-Centered Multimedia, HCM 2007*, pages 85 – 94. ACM New York, NY, USA, 2007. [Cited on page 232.]
- Charles Rich and Candace L. Sidner. COLLAGEN: A Collaboration Manager for Software Interface Agents. *Journal of User Modeling and User-Adapted Interaction*, 8(3-4):315–350, 1998. [Cited on pages 84, 87, 118, and 190.]
- Charles Rich and Candace L. Sidner. Using Collaborative Discourse Theory to Partially Automate Dialogue Tree Authoring. In *Proceedings of the 12th International Conference on Intelligent Virtual Agents, IVA 2012, Santa Cruz, California, USA, September 12-14, 2012*, pages 327–340. Springer-Verlag Berlin, Heidelberg, 2012. [Cited on pages 83, 84, 87, and 99.]
- Charles Rich, Candace L. Sidner, and Neal Lesh. Collagen: Applying Collaborative Discourse Theory to Human-computer Interaction. *Artificial Intelligence Magazine*, 22(4):15–25, October 2001. [Cited on pages 84 and 87.]
- Charles Rich, Neal Lesh, Andrew Garland, and Jeff Rickel. A Plug-in Architecture for Generating Collaborative Agent Responses. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2002, Bologna, Italy, July 15 - 19, 2002*, pages 782–789. ACM, New York, NY, USA, 2002. [Cited on page 84.]

- Charles Rich, Brett Ponsler, Aaron Holroyd, and Candace L. Sidner. Recognizing Engagement in Human-Robot Interaction. In *Proceedings of the 5th ACM/IEEE International Conference on Human-Robot Interaction, HRI 2010, Osaka, Japan, March 2-5, 2010*, pages 375–382, 2010. [Cited on pages 55, 76, 99, 172, 176, 177, and 180.]
- Daniel C. Richardson and Rick Dale. Looking to Understand - The Coupling between Speakers' and Listeners' Eye Movements and its Relationship to Discourse Comprehension. *Journal of Cognitive Science*, 29(6):1045–1060, November 2005. [Cited on page 34.]
- Michael J. Richardson, Kerry L. Marsh, and Richard C. Schmidt. Effects of Visual and Verbal Interaction on Unintentional Interpersonal Coordination. *Journal of Experimental Psychology: Human Perception and Performance*, 31(1):62–79, 2005. [Cited on pages 5, 16, 28, and 59.]
- Daniel C. Richardson, Rick Dale, and Natasha Z. Kirkham. The Art of Conversation Is Coordination: Common Ground and the Coupling of Eye Movements During Dialogue. *Journal of Psychological Science*, 18(5):407–413, May 2007. [Cited on pages 5 and 35.]
- Michael J. Richardson, Kerry L. Marsh, Robert W. Isenhower, Justin R. L. Goodman, and Richard C. Schmidt. Rocking Together: Dynamics of Intentional and Unintentional Interpersonal Coordination. *Journal of Human Movement Science*, 26(6):867–891, December 2007. [Cited on page 17.]
- Daniel C. Richardson, Rick Dale, and Kevin Shockley. Embodied Communication. In Ipke Wachsmuth, Manuela Lenzen, and Günther Knoblich, editors, *Synchrony and Swing in Conversation: Coordination, Temporal Dynamics and Communication*, pages 75–93. Oxford University Press, 2008. [Cited on pages 19 and 29.]
- Jeff Rickel and W. Lewis Johnson. Embodied Conversational Agents. In *Task-Oriented Collaboration with Embodied Agents in Virtual Worlds*, pages 95–122. MIT Press, Cambridge, MA, USA, 2000. [Cited on page 87.]
- Raoul Rickenberg and Byron Reeves. The Effects of Animated Characters on Anxiety, Task Performance, and Evaluations of User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2000, The Hague, Netherlands, April 1- 6, 2000*, pages 49–56. ACM, New York, NY, USA, 2000. [Cited on page 229.]
- Mark Riedl, C. J. Saretto, and R. Michael Young. Managing Interaction Between Users and Agents in a Multi-Agent Storytelling Environment. In *Proceedings of the 2nd Joint Conference on Autonomous Agents and Multi-Agent Systems, AAMAS 2003, Melbourne, Australia, July 14 - 18, 2003*, pages 741–748. ACM New York, NY, USA, 2003. [Cited on pages 84 and 231.]
- Gary Riley. C Language Integrated Production System (CLIPS). In *Encyclopedia of Computer Science and Technology*, volume 37. Marcel Dekker Inc., 1997. [Cited on pages 81 and 93.]
- Thomas Rist, Stephan Baldes, Patrick Gebhard, Michael Kipp, Martin Klesen, Peter Rist, and Markus Schmitt. CrossTalk: An Interactive Installation with Animated Presentation Agents. In Elisabeth Andre, Andy Clarke, Clive Fencott, Craig Lindley, Grethe Mitchell, and Frank Nack, editors, *Proceedings of the 2th Conference on Computational Semiotics for Games and New Media, COSIGN 2002, Universität Augsburg, Germany, September 2-4, 2002*, pages 61–67, 2002. [Cited on pages 85 and 199.]
- Thomas Rist, Elisabeth Andre, Stephan Baldes, Patrick Gebhard, Martin Klesen, Michael Kipp, Peter Rist, and Markus Schmitt. A Review on the Development of Embodied Presentation Agents and their Application Fields. In Helmut Prendinger and Mitsuro Ishizuka, editors, *Life-like Characters. Tools, Affective Functions and Applications*, Cognitive Technologies Series, pages 377–404. Springer-Verlag, Berlin, 2003. [Cited on pages 6, 15, and 199.]
- Hannes Ritschel and Elisabeth André. Real-Time Robot Personality Adaptation Based on Reinforcement Learning and Social Signals. In *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction, HRI 2017, Vienna, Austria, 2017*, pages 265–266. ACM, New York, NY, USA, 2017. [Cited on pages 238 and 246.]
- Tony Ro, Charlotte Russell, and Nilli Lavie. Changing Faces: A Detection Advantage in the Flicker Paradigm. *Journal of Psychological Science*, 12(1):94–99, January 2001. [Cited on page 37.]
- Laura F. Robinson and Harry T. Reis. The Effects of Interruption, Gender, and Status on Interpersonal Perceptions. *Journal of Nonverbal Behavior*, 13(3):141–153, September 1989. [Cited on page 173.]
- Derek B. Roger and Andrea Schumacher. Effects of Individual Differences on Dyadic Conversational Strategies. *Journal of Personality and Social Psychology*, 45(3):700–705, September 1983. [Cited on pages V and 41.]

- Derek Roger, Peter Bull, and Sally Smith. The Development of a Comprehensive System for Classifying Interruptions. *Journal of Language and Social Psychology*, 7(1):27–34, 1988. [Cited on pages 41, 42, and 173.]
- Sheena Rogers, Melanie Lunsford, Lars Strother, and Michael Kubovy. The Mona Lisa Effect: Perception of Gaze Direction in Real and Pictured Faces. In *Studies in Perception and Action VII*, pages 19–24. Lawrence Erlbaum, 2003. [Cited on page 74.]
- Raquel Ros, Séverin Lemaignan, E. Akin Sisbot, Rachid Alami, Jasmin Steinwender, Katharina Hamann, and Felix Warneken. Which One? Grounding the Referent Based on Efficient Human-Robot Interaction. In Carlo Alberto Avizzano and Emanuele Ruffaldi, editors, *Proceedings of the 19th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN, 2010, Principe di Piemonte - Viareggio, Italy, September 12-15, 2010*. IEEE, 2010. [Cited on pages 70, 76, 77, 99, and 162.]
- Howard M. Rosenfeld. Nonverbal Behavior and Communication. In Aron W. Siegman and Stanley Feldstein, editors, *Conversational Control Functions of Nonverbal Behavior*, pages 563–601. Hillsdale, NJ, England: Lawrence Erlbaum Associates, Inc., 1987. [Cited on page 22.]
- Kerstin Ruhland, Christopher E. Peters, Sean Andrist, Jeremy B. Badler, Norman I. Badler, Michael Gleicher, Bilge Mutlu, and Rachel McDonnell. A Review of Eye Gaze in Virtual Agents, Social Robotics and HCI: Behaviour Generation, User Interaction and Perception. *Journal Computer Graphics Forum, The Eurographics Association & John Wiley & Sons, Ltd. Chichester, UK*, 34(6):299–326, September 2015. [Cited on pages 15, 30, and 74.]
- Harvey Sacks, Emanuel Schegloff, and Gail Jefferson. A Simplest Systematics for the Organization of Turn-Taking. *Journal of Language*, 50(4):696–735, 1974. [Cited on pages 17, 21, 35, 36, 39, 41, 42, 43, 56, 60, 76, 80, 86, 172, 175, and 235.]
- Olof Sandgren, Richard Andersson, Joost van de Weijer, Kristina Hansson, and Birgitta Sahlén. Timing of Gazes in Child Dialogues: A Time-Course Analysis of Requests and Back-Channelling in Referential Communication. *International Journal of Language and Communication Disorders*, 47(4):373–383, July-August 2012. [Cited on page 36.]
- Maria Sapouna, Dieter Wolke, Natalie Vannini, Scott Watson, Sarah Woods, Wolfgang Schneider, Sibylle Enz, Lynne Hall, Ana Paiva, Elizabeth André, Kerstin Dautenhahn, and Ruth Aylett. Virtual Learning Intervention to Reduce Bullying Victimization in Primary School: A Controlled Trial. *Journal of Child Psychology and Psychiatry*, 51(1):104–112, January 2010. [Cited on page 233.]
- Albert E. Schefflen. The Significance of Posture in Communication Systems. *Psychiatry: Interpersonal and Biological Processes*, 27(3):316–331, November 1964. [Cited on page 18.]
- Emanuel A. Schegloff and Harvey Sacks. Opening up Closings. *Semiotica*, 8(4), January 1973. [Cited on page 84.]
- Emanuel A. Schegloff, Gail Jefferson, and Harvey Sacks. The Preference for Self-Correction in the Organization of Repair in Conversation. *Journal of Language*, 53(2):361–382, June 1977. [Cited on page 23.]
- Emanuel A. Schegloff. Sequencing in Conversational Openings. *American Anthropologist*, 70(6):1075–1095, December 1968. [Cited on pages 17, 36, 43, and 84.]
- Emanuel A. Schegloff. Between Micro and Macro: Contexts and Other Connections. In Jeffrey C. Alexander, Bernhard Giesen, Richard Munch, and Neil J. Smelser, editors, *The Micro-Macro Link*, page 207–234. University of California Press, Berkeley and Los Angeles, 1987. [Cited on page 40.]
- Emanuel A. Schegloff. Overlapping Talk and the Organization of Turn-Taking for Conversation. *Language in Society*, 29(1):1–63, March 2000. [Cited on pages 5, 35, 42, 43, 56, and 60.]
- Emanuel A. Schegloff. Accounts of Conduct in Interaction: Interruption, Overlap, and Turn-Taking. In Jonathan H. Turner, editor, *Handbook of Sociological Theory*, Handbooks of Sociology and Social Research, pages 287–321. Springer US, 2001. [Cited on pages 35, 42, 55, 56, and 60.]
- Stefan Scherer, Stacy Marsella, Giota Stratou, Yuyu Xu, Fabrizio Morbini, Alesia Egan, Albert Rizzo, and Louis-Philippe Morency. Perception Markup Language: Towards a Standardized Representation of Perceived Nonverbal Behaviors. In *Proceedings of the 12th International Conference on Intelligent Virtual Agents, IVA 2012, Santa Cruz, CA, USA, September, 12-14, 2012*, pages 455–463. Springer, 2012. [Cited on page 88.]
- Richard C. Schmidt and Michael J. Richardson. Dynamics of Interpersonal Coordination. In A. Fuchs and V. K. Jirsa, editors, *Coordination: Neural, Behavioral and Social Dynamics*, pages 281–307. Springer-Verlag, Berlin, 2008. [Cited on pages 15 and 28.]

- Richard C. Schmidt, Samantha Morr, Paula Fitzpatrick, and Michael J. Richardson. Measuring the Dynamics of Interactional Synchrony. *Journal of Nonverbal Behavior*, 36(4):263–279, December 2012. [Cited on pages 5, 16, 21, and 28.]
- Christophe Scholliers, Lode Hoste, Beat Signer, , and Wolfgang De Meuter. Midas: A Declarative Multi-Touch Interaction Framework. In *Proceedings of the , TEI 2011, Funchal, Portugal, January 22-26, 2011*, 2011. [Cited on page 82.]
- Marc Schröder, Patrick Gebhard, Marcela Charfuelan, Christoph Endres, Michael Kipp, Sathish Chandra Pammi, Martin Rumpler, and Oytun Türk. Enhancing Animated Agents in an Instrumented Poker Game. In Andreas Dengel, Karsten Berns, Thomas Breuel, Frank Bomarius, and Thomas Roth-Berghofer, editors, *Proceedings of the 31st Annual German Conference on Artificial Intelligence, KI 2008, Kaiserslautern, Germany, September 23-26, 2008*, volume 5243 of *Lecture Notes in Artificial Intelligence*, pages 316–323. Springer, 2008. [Cited on page 199.]
- Marc Schröder, Paolo Baggia, Felix Burkhardt, Catherine Pelachaud, Christian Peter, and Enrico Zovato. EmotionML - An Upcoming Standard for Representing Emotions and Related States. In Sidney D’Mello, Arthur Graesser, Björn Schuller, and Jean-Claude Martin, editors, *Proceedings of the 4th International Conference on Affective Computing and Intelligent Interaction, ACII 2011, Memphis, TN, USA, October 9-12, 2011*, volume 6974 of *Lecture Notes in Computer Science*, pages 316–325. Springer-Verlag Berlin Heidelberg, 2011. [Cited on pages 98 and 109.]
- Marc Schröder. Approaches to Emotional Expressivity in Synthetic Speech. In Krzysztof Izdebski, editor, *Emotions in the Human Voice: Culture and Perception*, volume 3, pages 307–321. Lural Publishing, San Diego, CA, USA, 2008. [Cited on page 10.]
- Andreas Schöter and Buccleuch Place. Compiling Feature Structures into Terms: An Empirical Study in Prolog. In *Recommendation ITU-T H.262 (MPEG 2), International Standard ISO/IEC*, pages 13818–2, 1993. [Cited on page 120.]
- Natalie Sebanz and Guenther Knoblich. Prediction in Joint Action: What, When, and Where. *Topics in Cognitive Science*, 1(2):353–367, April 2009. [Cited on pages 5, 33, 54, and 56.]
- Natalie Sebanz, Harold Bekkering, and Günther Knoblich. Joint Action: Bodies and Minds Moving Together. *Journal of Trends in Cognitive Sciences*, 10(2):70–76, February 2006. [Cited on pages 5, 27, 29, 34, 54, and 59.]
- Rohan Sharma. Toward Multimodal Human-Computer Interface. *Proceedings of the IEEE*, 86(5):853–869, May 1998. [Cited on page 63.]
- Stephen V. Shepherd. Following Gaze: Gaze-Following Behavior as a Window into Social Cognition. *Frontiers in Integrative Neuroscience*, 4(5), January 2010. [Cited on page 30.]
- James V. Sherwood. Facilitative Effects of Gaze upon Learning. *Perceptual and Motor Skills*, 64(3):1275–1278, June 1987. [Cited on page 35.]
- Stuart M. Shieber, Hans Uszkoreit, , Jane Robinson, and Mabry Tyson. Implementation of PATR-II. In Barbara J. Grosz and Mark E. Stickel, editors, *Research on Interactive Acquisition and Use of Knowledge. Final Report of SRI Project 1894*, pages 173–281. SRI International, Menlo Park, CA, USA, 1983. [Cited on page 119.]
- Stuart M. Shieber. The Design of a Computer Language for Linguistic Information. In *Proceedings of the 10th International Conference on Computational Linguistics and the 22nd Annual Meeting on Association for Computational Linguistics, ACL 1984, Stanford, CA, USA, July 5, 1984*, pages 362–366. Association for Computational Linguistics, Stroudsburg, PA, USA, 1984. [Cited on page 119.]
- Stuart M. Shieber. Using Restriction to Extend Parsing Algorithms for Complex-Feature-based Formalisms. In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics, ACL 1985, University of Chicago, Chicago, IL, USA, 8-12 July 1985*, pages 145–152. Association for Computational Linguistics, Stroudsburg, PA, USA, 1985. [Cited on page 119.]
- Stuart M. Shieber. *An Introduction to Unification-based Approaches to Grammar. Reissue of Shieber, Stuart M. 1986. An Introduction to Unification-based Approaches to Grammar. CSLI Publications, Stanford, CA, USA. Microtome Publishing, Brookline, MA, USA, 2003.* [Cited on page 119.]
- Kevin Shockley, Marie-Vee Santana, and Carol A. Fowler. Mutual Interpersonal Postural Constraints are Involved in Cooperative Conversation. *Journal of Experimental Psychology: Human Perception and Performance*, 29(2):326–32, April 2003. [Cited on page 18.]

- Elizabeth Shriberg, Andreas Stolcke, and Don Baron. Observations on Overlap: Findings and Implications for Automatic Processing of Multi-Party Conversation. In Paul Dalsgaard, Børge Lindberg, Henrik Benner, and Zheng-Hua Tan, editors, *Proceedings of the 7th European Conference on Speech Communication and Technology, EUROSPEECH 2001, Aalborg, Denmark, September 3 - 7, 2001*, pages 1359–1362, 2001. [Cited on pages 5, 17, 42, 43, and 44.]
- Mei Si, Stacy C. Marsella, and David V. Pynadath. Thespian: An Architecture for Interactive Pedagogical Drama. In *Proceedings of the 12th International Conference on Artificial Intelligence in Education, AIED 2005, Amsterdam, The Netherlands, July 18-22, 2005*, pages 595–602. IOS Press, 2005. [Cited on page 231.]
- Candace L. Sidner, Christopher Lee, Cory D. Kidd, Neal Lesh, and Charles Rich. Explorations in Engagement for Humans and Robots. *Journal of Artificial Intelligence*, 166(1-2):140–164, August 2005. [Cited on pages 34, 37, and 76.]
- Gabriel Skantze and Martin Johansson. Modelling Situated Human-Robot Interaction using IrisTK. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGDIAL 2015, Prague, Czech Republic, September 2-4, 2015*, pages 165–167. The Association for Computer Linguistics, 2015. [Cited on page 86.]
- Gabriel Skantze and Samer Al Moubayed. IrisTK: A Statechart-Based Toolkit for Multi-Party Face-To-Face Interaction. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction, ICMI 2012, Santa Monica, California, USA, October 22-26, 2012*, ACM International Conference Proceedings, pages 69–76. ACM, New York, NY, USA, 2012. [Cited on pages 86 and 133.]
- Gabriel Skantze, Anna Hjalmarsson, and Catharine Oertel. Turn-taking, Feedback and Joint Attention in Situated Human-Robot Interaction. *Speech Communication*, 65, 2014. [Cited on page 86.]
- Gabriel Skantze, Martin Johansson, and Jonas Beskow. Exploring Turn-taking Cues in Multi-party Human-robot Discussions about Objects. In Zhengyou Zhang, Phil Cohen, Dan Bohus, Radu Horaud, and Helen Meng, editors, *Proceedings of the 17th ACM on International Conference on Multimodal Interaction, ICMI 2015, Seattle, Washington, USA, November 9-13, 2015*, pages 67–74. ACM New York, NY, USA, 2015. [Cited on page 86.]
- Vicki L. Smith and Herbert H. Clark. On the Course of Answering Questions. *Journal of Memory and Language*, 32(1):25–38, February 1993. [Cited on page 25.]
- Justin S. Smith, Crystal Chao, and Andrea L. Thomaz. Real-Time Changes to Social Dynamics in Human-Robot Turn-Taking. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2015, Hamburg, Germany September 28 - October 2, 2015*, pages 3024–3029, 2015. [Cited on pages 80, 87, 99, and 184.]
- Lynn Smith-Lovin and Charles Brody. Interruptions in Group Discussions: The Effects of Gender and Group Composition. *American Sociological Review*, 54(3):424–435, June 1989. [Cited on pages 5, 42, and 44.]
- Gert Smolka and Ralf Treinen. Records for Logic Programming. *Journal of Logic Programming, Elsevier Science Publishing, New York, NY, USA*, 18(3):229–258, April 1994. [Cited on page 119.]
- Gert Smolka. Feature-Constraint Logics for Unification Grammars. *Journal of Logic Programming, Elsevier Science Publishing, New York, NY, USA*, 12(1-2):51–87, January 1992. [Cited on page 119.]
- Timo Sowa, Martin Fröhlich, and Marc Erich Latoschik. Temporal Symbolic Integration Applied to a Multimodal System Using Gestures and Speech. In Annelies Braffort, Rachid Gherbi, Sylvie Gibet, James Richardson, and Daniel Teil, editors, *Proceedings of the International Gesture Workshop: Gesture-Based Communication in Human-Computer Interaction, GW 1999, Gif-sur-Yvette, France, March 17-19, 1999*, Lecture Notes in Artificial Intelligence, pages 291–302. Springer-Verlag Berlin Heidelberg, 1999. [Cited on page 93.]
- Ulrike Spierling, Sebastian A. Weiß, and Wolfgang Müller. Towards Accessible Authoring Tools for Interactive Storytelling. In Stefan Göbel, Rainer Malkewitz, and Ido Iurgel, editors, *Proceedings of the 3rd International Conference on Technologies for Interactive Digital Storytelling and Entertainment, TIDSE 2006, Darmstadt, Germany, December 4-6, 2006*, volume 4326 of *Lecture Notes in Computer Science*, pages 169–180. Springer, Springer Berlin Heidelberg, 2006. [Cited on page 85.]
- Vasant Srinivasan and Robin R. Murphy. A Survey of Social Gaze. In *Proceedings of the 6th Annual Conference on Human-Robot Interaction, HRI 2011, Lausanne, Switzerland, March 6-9, 2011*, ACM New York, NY, USA, pages 253–254, 2011. [Cited on pages 15, 30, and 74.]

- Vasant Srinivasan, Cindy L. Bethel, and Robin R. Murphy. Evaluation of Head Gaze Loosely Synchronized With Real-Time Synthetic Speech for Social Robots. *IEEE Transactions on Human-Machine Systems*, 44(6):767–778, December 2014. [Cited on pages 74 and 193.]
- Gordon Stanley and Donald S. Martin. Eye-Contact and the Recall of Material Involving Competitive and Non-competitive Associations. *Journal of Psychonomic Science: Human Learning and Thinking Social Processes*, 13(6):337–338, 1968. [Cited on page 37.]
- Maria Staudte and Matthew W. Crocker. Visual Attention in Spoken Human-Robot Interaction. In *Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction, HRI 2009, La Jolla, California, USA, March 11-13, 2009*, pages 77–84. ACM, New York, NY, USA, 2009. [Cited on pages 74 and 77.]
- Maria Staudte and Matthew W. Crocker. Investigating Joint Attention Mechanisms Through Spoken Human-Robot Interaction. *Journal of Cognition*, 120(2):268–291, August 2011. [Cited on pages 74, 76, 77, 99, and 235.]
- Maria Staudte. *Joint Attention in Spoken Human-Robot Interaction*. PhD thesis, Saarland University, 2010. [Cited on page 77.]
- Anna-Brita Stenstrom, editor. *An Introduction to Spoken Interaction*. London: Longman Group UK Limited, 1994. [Cited on page 43.]
- Daniel Stern. Mother and Infant at Play: The Dyadic Interaction Involving Facial, Vocal and Gaze Behaviors. In M. Lewis and L. Rosenblum, editors, *The Effect of the Infant on its Caregiver*, pages 187–213. New York: Wiley, 1974. [Cited on page 19.]
- Rainer Stiefelhagen, Christian Fügen, Petra Gieselmann, Hartwig Holzapfel, Kai Nickel, and Alex Waibel. Natural Human-Robot Interaction using Speech, Head Pose and Gestures. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2004, Sendai, Japan, September 28 - October 2, 2004*, pages 1–6, 2004. [Cited on pages 92, 93, and 119.]
- Walter Dan Stiehl, Jeff Lieberman, Cynthia Breazeal, Louis Basel, Levi Lalla, and Michael Wolf. The Design of the Huggable: A Therapeutic Robotic Companion for Relational, Affective Touch. In *Proceedings of the AAAI Symposium on Caring Machines: AI in Eldercare, Washington D. C., USA, November 3-6, 2006*, pages 1–8, 2006. [Cited on page 6.]
- Brian A. Stone and James C. Lester. Dynamically Sequencing an Animated Pedagogical Agent. In *Proceedings of the 13-th National Conference on Artificial Intelligence, AAAI 1996, Portland, Oregon, 1996*, pages 424–431, 1996. [Cited on page 98.]
- Martin Strauss and Michael Kipp. ERIC: A Generic Rule-Based Framework for an Affective Embodied Commentary Agent. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems, Estoril, Portugal, May 12-15, 2008*, pages 97–104. International Foundation for Autonomous Agents and Multiagent Systems, 2008. [Cited on page 82.]
- Yong Sun, Yu Shi, Fang Chen, and Vera Chung. An Efficient Unification-Based Multimodal Language Processor in Multimodal Input Fusion. In *Proceedings of the 19th Australasian Conference on Computer-Human Interaction, OZCHI 2007, Entertaining User Interfaces, Adelaide, Australia, November 28-30, 2007*, pages 215–218, 2007. [Cited on pages 92 and 119.]
- Yong Sun, Yu Shi, Fang Chen, and Vera Chung. Building a Practical Multimodal System with a Multimodal Fusion Module. In Julie A. Jacko, editor, *Proceedings of the 13th International Conference on Human-Computer Interaction- Novel Interaction Methods and Techniques, HCI International 2009, San Diego, CA, USA, July 19-24, 2009*, pages 93–102. Springer Berlin Heidelberg, 2009. [Cited on page 92.]
- Stephen Sutton and Ronald Cole. The CSLU Toolkit: Rapid Prototyping of Spoken Language Systems. In *Proceedings of the 10th Annual Symposium on User Interface Software and Technology, UIST 1997, Banff, Alberta, Canada, October 14-17, 1997*, pages 85–86. ACM New York, NY, USA, 1997. [Cited on page 85.]
- W. Swartout, J. Gratch, W.L. Johnson, C. Kyriakakis, C. LaBore, R. Lindheim, S. Marsella, D. Miraglia, B. Moore, J. Morie, J. Rickel, M. ThiÚbaux, L. Tuch, R. Whitney, and J. Douglas. Toward the Holodeck: Integrating Graphics, Sound, Character and Story. In *Proceedings of the 5th International Conference on Autonomous Agents, AGENTS 2001, Montreal, Quebec, Canada, May 28 - June 1, 2001*, pages 409–416. ACM, New York, NY, USA, 2001. [Cited on page 87.]

- William R. Swartout, Jonathan Gratch, Randall W. Hill Jr., Eduard Hovy, Stacy Marsella, Jeff Rickel, and David Traum. Toward Virtual Humans. *AI Magazine, Association for the Advancement of Artificial Intelligence*, 27(2):96–108, 2006. [Cited on pages 87, 99, and 231.]
- Nicolas Szilas. IDtension: A Narrative Engine for Interactive Drama. In *Proceedings of the 1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment, TIDSE 2003, Darmstadt, Germany, March 24–26, 2003*, pages 1–11, 2003. [Cited on page 84.]
- Michael K. Tanenhaus and John C. Trueswell. Sentence Comprehension. In Joanne L. Miller and Peter D. Eimas, editors, *Speech, Language, and Communication. Handbook of Perception and Cognition*, pages 217–262. Academic Press, San Diego, CA, USA, 1995. [Cited on page 23.]
- Deborah Tannen. *Conversational Style: Analyzing Talk Among Friends*. Norwood, N.J.: Ablex, 1984. [Cited on pages 5, 41, 43, and 44.]
- Deborah Tannen. *Gender and Discourse*. Oxford University Press, 1994. [Cited on pages 5, 15, 41, 42, 43, 56, 60, and 173.]
- Deborah Tannen. Turn-taking and Intercultural Discourse and Communication. In Christina Paulston, Scott Kiesling, and Elizabeth Rangel, editors, *The Handbook of Intercultural Discourse and Communication*, pages 135–157. Chichester, UK: John Wiley and Sons, 2012. [Cited on pages 15, 39, 41, and 43.]
- Paul Tarau. Jinni: A Lightweight Java-Based Logic Engine for Internet Programming. In K. Sagonas, editor, *Proceedings of the International Workshop on Implementation Technologies for Programming Languages based on Logic, JICSLP 1998, Manchester, UK, 1998*, pages 1–15, 1998. [Cited on page 81.]
- Mark ter Maat and Dirk Heylen. Turn Management or Impression Management? In Zsófia Ruttkay, Michael Kipp, Anton Nijholt, and Hannes Högni Vilhjálmsson, editors, *Proceedings of the 9th International Conference on Intelligent Virtual Agents, IVA 2009, Amsterdam, The Netherlands, September 14–16, 2009*, volume 5773 of *Lecture Notes in Computer Science*, pages 467–473. Springer-Verlag, Berlin, Heidelberg, 2009. [Cited on page 185.]
- Mark ter Maat, Khiet P. Truong, and Dirk Heylen. How Turn-Taking Strategies Influence Users' Impressions of an Agent. In Jan M. Allbeck, Norman I. Badler, Timothy W. Bickmore, Catherine Pelachaud, and Alla Safonova, editors, *Proceedings of the 10th International Conference on Intelligent Virtual Agents, IVA 2010, Philadelphia, Pennsylvania, USA, September 20–22, 2010*, volume 6356 of *Lecture Notes in Computer Science*, pages 441–453. Springer-Verlag, Berlin, Heidelberg, 2010. [Cited on pages 173 and 185.]
- Mark ter Maat, Khiet P. Truong, and Dirk Heylen. How Agents' Turn-Taking Strategies Influence Impressions and Response Behaviors. *Presence: Teleoperators and Virtual Environments*, 20(5):412–430, October 2011. [Cited on pages 173, 184, and 185.]
- Andrea L. Thomaz and Crystal Chao. Turn-Taking Based on Information Flow for Fluent Human-Robot Interaction. *AI Magazine: Special Issue on Dialog with Robots*, 32(4):53–63, 2011. [Cited on pages 80 and 87.]
- Craig W. Thompson. Building Menu-Based Natural Language Interfaces. *Texas Engineering Journal*, 3:140–150., 1986. [Cited on page 89.]
- Kristinn R. Thórisson, Olafur Gislason, Gudny Ragna Jonsdottir, and Hrafn Th. Thorisson. A Multiparty Multimodal Architecture for Realtime Turntaking. In Jan M. Allbeck, Norman I. Badler, Timothy W. Bickmore, Catherine Pelachaud, and Alla Safonova, editors, *Proceedings of the 10th International Conference on Intelligent Virtual Agent, IVA 2010, Philadelphia, Pennsylvania, USA, September 20–22, 2010*, volume 6356 of *Lecture Notes in Computer Science*, pages 350–356. Springer-Verlag, Berlin, Heidelberg, 2010. [Cited on pages 80 and 170.]
- Kristinn R. Thórisson. Natural Turn-Taking Needs No Manual: Computational Theory and Model, from Perception to Action. In Björn Granström, David House, and Inger Karlsson, editors, *Multimodality in Language and Speech Systems*, volume 19 of *Text, Speech and Language Technology*, pages 173–207. Springer Netherlands, 2002. [Cited on pages 80 and 170.]
- Michael Tomasello, Malinda Carpenter, Josep Call, and Tanya Behne. Understanding and Sharing Intentions: The Origins of Cultural Cognition. *Journal of Behavioral and Brain Sciences*, 28(5):675–669, October 2005. [Cited on page 31.]
- Michael Tomasello. Joint Attention as Social Cognition. In Chris Moore and Phil Dunham, editors, *Joint Attention: Its Origins and Role in Development*, pages 103–130. Psychology Press, New York, NY, USA, 1995. [Cited on pages 31, 33, and 54.]

- David R. Traum and Staffan Larsson. The Information State Approach to Dialogue Management. In Jan van Kuppevelt and Ronnie W. Smith, editors, *Current and New Directions in Discourse and Dialogue*, volume 22 of *Text, Speech and Language Technology*, pages 325–353. Springer Netherlands, 2003. [Cited on page 86.]
- David Traum and Jeff Rickel. Embodied Agents for Multi-Party Dialogue in Immersive Virtual Worlds. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2002, Bologna, Italy, July 15 -19, 2002*, pages 766–773. ACM, New York, NY, USA, 2002. [Cited on pages 77 and 170.]
- David Traum, Anton Leuski, Antonio Roque, Sudeep Gandhe, David DeVault, Jillian Gerten, Susan Robinson, and Bilyana Martinovski. Natural Language Dialogue Architectures for Tactical Questioning Characters. In *Proceedings of the 26th Army Science Conference, Orlando, Florida, USA, December 1-4, 2008*, pages 1–7. Fort Belvoir, Defense Technical Information Center, DEC, 2008. [Cited on pages 6, 10, and 86.]
- David Traum. *A Computational Theory of Grounding in Natural Language Conversation*. PhD thesis, Department of Computer Science, University of Rochester, 1994. [Cited on page 70.]
- Deborah L. Trout and Howard M. Rosenfeld. The Effect of Postural Lean and Body Congruence on the judgement of Psychotherapeutic Rapport. *Journal of Nonverbal Behavior*, 4(3):176–190, March 1980. [Cited on page 19.]
- H. Trung. Multimodal Dialogue Management - State-of-the-Art. *Human Media Interaction Department, University of Twente*, 2, 2006. [Cited on page 81.]
- Wolfgang Tschacher, Georg M. Rees, and Fabian Ramseyer. Nonverbal Synchrony and Affect in Dyadic Interactions. *Frontiers in Psychology*, 5(1323):1–13, November 2014. [Cited on page 21.]
- Matthew Turk. Multimodal Interaction: A Review. *Pattern Recognition Letters*, 36(15):189–195, January 2014. [Cited on page 89.]
- Stefan Ultes and Wolfgang Minker. Managing Adaptive Spoken Dialogue for Intelligent Environments. *Journal of Ambient Intelligence and Smart Environments*, 6:523–539, 2014. [Cited on pages 10, 190, and 239.]
- Stefan Ultes, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Iñigo Casanueva, Paweł Budzianowski, Nikola Mrkvić, Tsung-Hsien Wen, Milica Gašić, , and Steve J. Young. PyDial: A Multi-Domain Statistical Dialogue System Toolkit. In *In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, System Demonstrations, Vancouver, Canada, July 30- August 4, 2017*, pages 73–78. Association of Computational Linguistics, 2017. [Cited on page 84.]
- Piercarlo Valdesolo and David DeSteno. Synchrony and the Social Tuning of Compassion. *Journal of Emotion*, 11(2):262–266, April 2011. [Cited on pages 21 and 29.]
- Rick van Baaren, Loes Janssen, Tanya L. Chartrand, and Ap Dijksterhuis. Where is the Love? The Social Aspects of Mimicry. *Philosophical Transactions of the Royal Society*, 364:2381–2389, August 2009. [Cited on page 18.]
- Kees van Deemter, Emiel Krahmer, and Mariët Theune. Real versus Template-Based Natural Language Generation: A False Opposition? In *Journal of Computational Linguistics, MIT Press Cambridge, MA, USA*, pages 15–24, March 2005. [Cited on pages 71 and 98.]
- Arie van Deursen, Paul Klint, and Joost Visser. Domain-Specific Languages: An Annotated Bibliography. *ACM SIGPLAN Notices, ACM, New York, NY, USA*, 35(6):26–36, June 2000. [Cited on pages 67 and 104.]
- Susanne van Mulken, Elisabeth André, and Jochen Müller. The Persona Effect: How Substantial Is It? In Hilary Johnson, Lawrence Nigay, and Christopher Roast, editors, *Proceedings of HCI on People and Computers XIII*, pages 53–66, London, UK, 1998. Springer London. [Cited on page 230.]
- H. van Welbergen, D. Reidsma, and Stefan Kopp. An Incremental Multimodal Realizer for Behavior Co-Articulation and Coordination. In *In Proceedings of the 12th International Conference on Intelligent Virtual Agents, IVA 2012, Santa Cruz, CA, USA, 2012*, pages 175–188, 2012. [Cited on pages 98 and 99.]
- Ivar E. Vermeulen, Christian Roth, Peter Vorderer, and Christoph Klimmt. Measuring User Responses to Interactive Stories: Towards a Standardized Assessment Tool. In Ruth Aylett, Mei Yui Lim, Sandy Louchart, Paolo Petta, and Mark Riedl, editors, *Proceedings of the 3rd Joint Conference on Interactive Digital Storytelling, ICIDS 2010, Edinburgh, UK, November 1-3, 2010*, volume 6432 of *Lecture Notes in Computer Science*, pages 38–43. Springer Berlin Heidelberg, 2010. [Cited on page 233.]
- Roel Vertegaal, Gerrit van der Veer, and Harro Vons. Effects of Gaze on Multiparty Mediated Communication. In *Proceedings of the Graphics Interface, GI 2000, Montreal, Quebec, Canada, May 15-17, 2000*, pages 95–102, 2000. [Cited on page 35.]

- Hannes Vilhjálmsón, Nathan Cantelmo, Justine Cassell, Nicolas E. Chafai, Michael Kipp, Stefan Kopp, Maurizio Mancini, Stacy Marsella, Andrew N. Marshall, Catherine Pelachaud, Zsofi Ruttkay, Kristinn R. Thórisson, Herwin van Welbergen, and Rick J. van der Werf. The Behavior Markup Language: Recent Developments and Challenges. In Catherine Pelachaud, Jean-Claude Martin, Elisabeth André, Gérard Chollet, Kostas Karpouzis, and Danielle Pelé, editors, *Proceedings of the 7th International Conference on Intelligent Virtual Agents, IVA 2007, Paris, France, September 17-19, 2007*, volume 4722 of *Lecture Notes in Computer Science*, pages 99–111. Springer-Verlag Berlin Heidelberg, 2007. [Cited on pages 88, 97, 98, and 109.]
- Vinoba Vinayagamoorthy, Marco Gillies, Anthony Steed, Emmanuel Tanguy, Xueni Pan, Celine Loscos, and Mel Slater. Building Expression into Virtual Characters. In Brian Wyvill and Alexander Wilkie, editors, *Proceedings of the annual main conference of the European Association for Computer Graphics, Eurographics 2006, State of the Art Reports, Vienna, Austria, September 4-8, 2006*, 2006. [Cited on pages 10 and 68.]
- Minh Tue Vo and Alex Waibel. Modeling and Interpreting Multimodal Inputs: A Semantic Integration Approach. Technical report, Carnegie-Mellon University Pittsburgh School of Computer Science, 1997. [Cited on page 91.]
- Minh Tue Vo and Cindy Wood. Building an Application Framework for Speech and Pen Input Integration in Multimodal Learning Interfaces. In *International Conference on Acoustics, Speech, and Signal Processing, ICASSP 1996, Atlanta, Georgia, USA, May 07 - 10, 1996*, pages 3545–3548. IEEE Computer Society Washington, DC, USA, 1996. [Cited on page 91.]
- Mario von Cranach and Johann H. Ellgring. Problems in the Recognition of Gaze Direction. In *Social Communication and Movement: Studies of Interaction and Expression in Man and Chimpanzee*. Academic Press, London, 1973. [Cited on pages 30, 31, and 32.]
- Mario von Cranach and Johann H. Ellgring. The Perception of Looking Behaviour. In M. von Cranach and I. Vine, editors, *Social Communication and Movement*, pages 419–443. Academic Press, London, 1973. [Cited on pages 30 and 31.]
- Michael von der Beeck. A Comparison of Statecharts Variants. In Hans Langmaack, Willem-Paul de Roever, and Jan Vytöpil, editors, *Formal Techniques in Real-Time and Fault-Tolerant Systems, Proceedings of the 3rd International Symposium Organized Jointly with the Working Group Provably Correct Systems, ProCoS 1994, Lübeck, Germany, September 19-23, 1994*, pages 128–148. Springer-Verlag Berlin-Heidelberg, 1994. [Cited on pages 56, 86, 201, and 205.]
- Aldert Vrij. Deception in Children: A Literature Review and Implications for Children’s Testimony. In H. L. Westcott, G. M. Davies, and R. H. C. Bull, editors, *Children’s Testimony: A Handbook of Psychological Research and Forensic Practice*, pages 175–194. Wiley, London, UK, 2002. [Cited on page 37.]
- Johannes Wagner, Florian Lingens, Tobias Baur, Ionut Damian, Felix Kistler, and Elisabeth André. The Social Signal Interpretation (SSI) Framework: Multimodal Signal Processing and Recognition in Real-time. In *Proceedings of the 21st ACM International Conference on Multimedia, MM 2013, Barcelona, Catalunya, Spain, October 21-25, 2013*, pages 831–834. ACM, New York, NY, USA, 2013. [Cited on pages VI, 10, 123, 152, 155, 156, and 239.]
- Wolfgang Wahlster. *SmartKom: Foundations of Multimodal Dialogue Systems*. Cognitive Technologies. SmartKom: Foundations of Multimodal Dialogue Systems, 2006. [Cited on page 69.]
- Leo Wanner, Josep Blat, Stamati Dasiopoulou, Mónica Domínguez, Gerard Llorach, Simon Mille, Federico Sunko, Eleni Kamateri, Stefanos Vrochidis, Ioannis Kompatsiaris, Elisabeth André, Florian Lingens, Gregor Mehlmann, Andries Stam, Ludo Stellingwerff, Bianca Vieru, Lori Lamel, Wolfgang Minker, Louisa Pragst, and Stefan Ultes. Towards a Multimedia Knowledge-Based Agent with Social Competence and Human Interaction Capabilities. In *Proceedings of the 1st International Workshop on Multimedia Analysis and Retrieval for Multimodal Interaction, MARMI 2016, New York, USA, June 6, 2016*, pages 21–26. ACM, New York, NY, USA, 2016. [Cited on pages 151 and 239.]
- Nigel G. Ward, David, and DeVault. Ten Challenges in Highly-Interactive Dialog Systems. In *AAAI Spring Symposium on Turn-Taking and Coordination in Human-Machine Interaction*, pages 104–107. Association for the Advancement of Artificial Intelligence, 2015. [Cited on page 173.]
- Rebecca M. Warner, Daniel Malloy, Kathy Schneider, Russell Knoth, and Bruce Wilder. Rhythmic Organization of Social Interaction and Observer Ratings of Positive Affect and Involvement. *Journal of Nonverbal Behavior*, 11(2):57–74, June 1987. [Cited on page 22.]
- David H. D. Warren and Fernando C. N. Pereira. An Efficient Easily Adaptable System for Interpreting Natural Language Queries. *Journal of Computational Linguistics*, 8(3-4):110–122, July 1982. [Cited on page 130.]

- Rainer Wasinger, Antonio Krüger, and Oliver Jacobs. Integrating Intra and Extra Gestures into a Mobile and Multimodal Shopping Assistant. In *Proceedings of the 3rd International Conference on Pervasive Computing, PERVASIVE 2005, Munich, Germany, May 8-13, 2005*, pages 297–314. Springer-Verlag Berlin Heidelberg, 2005. [Cited on pages 90, 91, and 119.]
- Rainer Wasinger. *Multimodal Interaction with Mobile Devices: Fusing a Broad Spectrum of Modality Combinations*, volume 305 of *Dissertations in Artificial Intelligence*. Akademische Verlagsgesellschaft Aka GmbH, Berlin, December 2006. [Cited on pages 90, 91, and 119.]
- Anthony I. Wasserman. Extending State Transition Diagrams for the Specification of Human-Computer Interaction. *IEEE Transactions on Software Engineering - Annals of discrete mathematics, IEEE Press Piscataway, NJ, USA*, 11(8):699–713, August 1985. [Cited on pages 95 and 133.]
- Eric D. Wesselmann, Florencia D. Cardoso, Samantha Slater, and Kipling D. Williams. To Be Looked at as Through Air: Civil Attention Matters. *Journal of Psychological Science*, 23(2):166–168, February 2012. [Cited on page 36.]
- Candace West and Don H. Zimmerman. Small Insults: A Study of Interruptions in Cross-Sex Conversations between Unacquainted Persons. In Barry Thorne, Cheris Kramarae, and Nancy Henley, editors, *Language, Gender and Society*, pages 102–117. Newbury House, Rowley, MA, 1983. [Cited on pages 40 and 44.]
- Jan Wielemaker, Tom Schrijvers, Markus Triska, and Torbjörn Lager. SWI-Prolog. *Theory and Practice of Logic Programming*, 12(1-2):67–96, 2012. [Cited on pages 67, 81, 93, and 122.]
- Arthur N. Wiens, Joseph D. Matarazzo, George Saslow, Shirley M. Thompson, and Ruth G. Matarazzo. Interview Interaction Behavior of Supervisors, Head Nurses, and Staff Nurses. *Journal of Nursing Research*, 14(4):322–329, Fall 1965. [Cited on page 40.]
- Preben Wik, Anna Hjalmarsson, and Jenny Brusk. DEAL: A Serious Game For CALL Practicing Conversational Skills In The Trade Domain. In *Proceedings of the Workshop on Speech and Language Technology in Education, SLATE 2007, Farmington, Pennsylvania, USA, October 1-3, 2007*, pages 88–91, 2007. [Cited on page 86.]
- Deanna Wilkes-Gibbs and Herbert H. Clark. Coordinating Beliefs in Conversation. *Journal of Memory and Language*, 13(2):183–194, April 1992. [Cited on pages 23, 33, and 183.]
- Kipling D. Williams. *Ostracism: The Power of Silence*. The Guilford Publications, New York, NY, USA, 2001. [Cited on page 36.]
- Scott S. Wiltermuth and Chip Heath. Synchrony and Cooperation. *Journal of Psychological Science*, 20(1):1–5, January 2009. [Cited on pages 21 and 29.]
- Niklaus Wirth. What Can We Do About the Unnecessary Diversity of Notation for Syntactic Definitions? *Communications of the ACM*, 20(11):822–823, 1977. [Cited on page 109.]
- Michael Wißner, Wouter Beek, Esther Lozano, Gregor U. Mehlmann, Floris Linnebank, Jochem Liem, Markus Häring, René Bühling, Jorge Gracia, Bert Bredeweg, and Elisabeth André. Character Roles and Interaction in the DynaLearn Intelligent Learning Environment. In *Proceedings of the 15th International Conference on Artificial Intelligence in Education, AIED 2011, Auckland, New Zealand, June 28 - July 2011*, volume 6738 of *Lecture Notes in Computer Science*, pages 585–587. Springer-Verlag, Berlin, Heidelberg, 2011. [Cited on page 231.]
- Michael Wißner, Wouter Beek, Esther Lozano, Gregor U. Mehlmann, Floris Linnebank, Jochem Liem, Markus Häring, René Bühling, Jorge Gracia, Bert Bredeweg, and Elisabeth André. Increasing Learners' Motivation through Pedagogical Agents: The Cast of Virtual Characters in the DynaLearn ILE. In Martin Beer, Cyril Brom, Frank Dignum, and Von-Wun Soo, editors, *Proceedings of the International Workshop on Agents for Educational Games and Simulations, AEGS 2011, Taipei, Taiwan, May 2, 2011*, volume 7471 of *Lecture Notes in Computer Science*, pages 151–165. Springer-Verlag, Berlin, Heidelberg, 2012. [Cited on page 231.]
- Anita E. Woolfolk and Douglas M. Brooks. The Influence of Teachers' Nonverbal Behaviors on Students' Perceptions and Performance. *The Elementary School Journal*, 85(4):513–528, March 1985. [Cited on page 35.]
- Lizhong Wu, Sharon L. Oviatt, and Philip R. Cohen. Multimodal Integration - A Statistical View. *IEEE Transactions on Multimedia*, 1(4):334–341, December 1999. [Cited on pages 91, 92, and 119.]
- Lizhong Wu, Sharon L. Oviatt, and Philip R. Cohen. From Members to Teams to Committee - A Robust Approach to Gestural and Multimodal Recognition. *IEEE Transactions on Neural Networks*, 13(4):972–982, July 2002. [Cited on page 92.]

- Wei Xu and Alexander I. Rudnicky. Task-Based Dialog Management using an Agenda. In *Proceedings of the ANLP/NAACL-ConvSys 2000 Workshop on Conversational Systems, Seattle, Washington, USA, May 4, 2000*, 2000. [Cited on page 84.]
- Songhua Xu, Hao Jiang, and Francis Lau. Personalized Online Document, Image and Video Recommendation via Commodity Eye-Tracking. In *Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys 2008, Lousanne, Switzerland, October 23-25, 2008*, pages 83–90. ACM New York, NY, USA, 2008. [Cited on pages 66 and 159.]
- Akiko Yamazaki, Keiichi Yamazaki, Yoshinori Kuno, Matthew Burdelski, Michie Kawashima, and Hideaki Kuzuoka. Precision Timing in Human-Robot Interaction: Coordination of Head Movement and Utterance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2008, Florence, Italy, April 5 - 10, 2008*, pages 131–140. ACM, New York, NY, USA, 2008. [Cited on page 76.]
- Stephen Yantis and John Jonides. Abrupt Visual Onsets and Selective Attention: Voluntary Versus Automatic Allocation. *Journal of Experimental Psychology: Human Perception and Performance*, 16(1):121–134, February 1990. [Cited on page 37.]
- Steven Yantis. Stimulus-Driven Attentional Capture and Attentional Control Settings. *Journal of Experimental Psychology: Human Perception and Performance*, 19(3):676–681, June 1993. [Cited on page 37.]
- Victor H. Yngve. On Getting a Word in Edgewise. *Papers from the 6th Regional Meeting of the Chicago Linguistic Society*, pages 657–677, 1970. [Cited on pages 5, 17, 25, 36, 37, 43, 54, 68, 179, and 235.]
- Jeff Youngquist. The Effect of Interruptions and Dyad Gender Combination on Perceptions of Interpersonal Dominance. *Journal of Communication Studies*, 60(2):147–163, April 2009. [Cited on pages 5 and 44.]
- Don H. Zimmerman and Candace West. Sex Roles, Interruptions and Silences in Conversation. In Barrie Thorne and Nancy Henley, editors, *Language and Sex: Difference and Dominance*, pages 105–129. Rowley, MA: Newbury House, 1975. [Cited on pages 5, 39, 43, 44, 187, and 195.]
- Miron Zuckerman and Robert E. Driver. Telling Lies: Verbal and Nonverbal Correlates of Deception. In A. W. Siegman and S. Feldstein, editors, *Multichannel Integrations of Nonverbal Behavior*. Erlbaum, Hillsdale, NJ, USA, 1985. [Cited on page 37.]