See, hear, listen: Head movements and the Sensitive Artificial Listener

Ruben Kooijman

Master's thesis, November 2007

committee:

Dirk Heylen

Rieks op den Akker

Anton Nijholt

Mannes Poel

Human Media Interaction

University of Twente
Enschede - The Netherlands

Abstract

This thesis looks at several aspects of head movements used in dyadic face-to-face conversations. Head movement behaviors are analyzed on external appearance (form) and how they are used, their meanings and purposes in conversations (function). Knowledge gained from these analyses can be incorporated in computational models for a Sensitive Artificial Listener (SAL) agent.

Video data is used for an analyses of head movement forms and functions. It is shown in this thesis that a rich but tangible description of head movement form, here called elementary head movement (EM), may contribute to automatic perception and synthesis of head movements. Furthermore it is investigated how head movement form can be mapped to head movement function in specific contexts. Special attention is given to head movement and gaze patterns in interactive contexts such as floor transitions and back-channels. The IAM framework presented at the end of in this work, enables rule-like language constructs for describing the internal processes of SAL. A prototype system incorporating the ideas of IAM is implemented, and rules defined in the prototype system, based on models from literature are defined in IAM to automatically produce video output of listener behavior.

Acknowledgements

Doing research on listening and head movements in conversations has not been that boring during the last two years. Just watching the videos of some friends and acquaintances talking to each other, making some annotations, philosophizing about it, fixing the head tracker et cetera, et cetera, no problem. But then, in the end some results are expected in the form of written reports and eventually the thesis... This was just too paradoxal for me at some times. “Where’s the listening, head movements, interaction and conversation in writing stuff down?”, I asked myself. Nonetheless, I knew this was what had to be done. Also, there were a lot of distractions in the past two years, some good some not so good. In the end something came out, this is it.

I really like to thank everybody I met at HMI for the shared ideas, inspirations, and serious support. Especially, I want to thank my supervisor Dirk Heylen for his inexhaustable kindness, his patience, the inspiration he gave me, and his trust.

Furthermore thanks go out to my parents, Ans en Kees, for their support, especially at difficult times. Dieuwke, my girlfriend, had to put up with me, my endless bla-bla, and her own health at times, but still supported me all the way, unbelievable. Now I can do nothing but hope I can pay you back for the support and patience. I’ll try to keep quiet now and listen.

Ruben Kooijman, 28-11-2007, Enschede.

Contents

Chapter 1

Introduction

1.1 Background

Some researchers in the area of human-computer interaction (HCI) have convincingly shown that people interact with computers as if they were human [Rosalind W. Picard, 1997]. But, as most computer programs do not have many human characteristics, people tend to get frustrated, or held back. One way to relieve this is to incorporate useful human characteristics in user interfaces. This is useful in virtual characters for e.g. entertainment purposes, and in embodied conversational agents that serve as tutors, receptionists, sales agents, or assistants so as to make these agent more convincing and believable. In chat sessions or virtual meeting rooms avatars can also make use of these characteristics to autonomously take over some natural behavior of their human counterpart to reduce bandwidth, or sensory processing and equipment.

In conversation people use not only verbal means of communication, but also para-verbal and nonverbal means. Virtual speakers, such as presentation agents, make use of these means to better convey the message to be communicated. But, as interaction involves at least two participants, listener behavior should also be taken into account. Listeners however, because they are non-speakers, rely more on non-verbal and para-verbal signals if they want to express themselves. So how do listeners express themselves, and how do they react to speaker behavior? And, what does an artificial character need to be a believable, convincing listener?

1.2 Goals

In this project, focus is on head movements. Several aspects of head movement behavior are highlighted to enable a Sensitive Artificial Listener (SAL) to exhibit real-time responsive conversational behavior. These aspects range from automatic perception of head movements and other behaviors to the modelling and synthesis of listener behavior. The first goal is to give an overview of the whole perception, action loop and explore what schemas and models are used in earlier work and what can be achieved by them. The second goal to use the insights gained by these explorations to propose a unified way of getting grip on the variations and complexities human conversation naturally brings about. The findings in this report can then contribute to the construction of a SAL. Therefore some models and frameworks will be implemented in prototype systems and shortly evaluated.

1.3 Approach

A basic method for developing models for Embodied Conversational Agents practiced by researchers such as Justine Cassell [2000], is to take real human behavior as a starting point. Through observation and analysis computational models can be defined, implemented and evaluated. Figure 1.1 depicts this process that will function as a guide throughout the report.

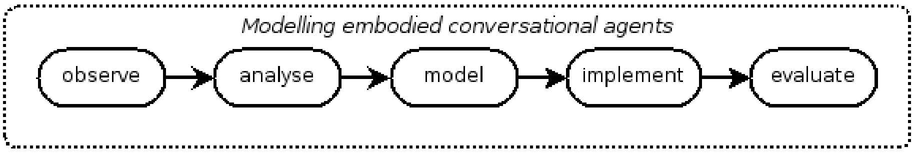


Figure 1.1: developing models for ECAs

In this report the following aspects of head movements during face-to-face conversations are explored:

- externals of head movements: what do they look like, how can they be transcribed?
- interpreting speaker behavior: what head movements are made by a speaker in specific contexts during the conversation?
- analysis of interactional behavior: how do listeners and speakers interact, which behavior patterns are involved?
- modelling and synthesis of artificial listener behavior: how can we design a listener agent?

It is assumed here that the above aspects are crucial pieces in the puzzle of creating an ECA capable of using human-like head movements as an artificial listener. In this thesis we make exploratory incursions in the aspects mentioned above.

1.4 Structure of the report

The report will start by investigating the movements themselves in chapter 2. Here answers are given to the questions like: what kind of head movements are studied, what are their properties, how can a computer program see what's happening?

Next the movements are placed in the context of face-to-face dyadic conversations in chapters 3 and 4. Here some real, video taped conversations are observed and qualitatively analyzed. A connection is made between theories and earlier findings in literature and the research performed here.

The last chapter (chapter 5) deals with a proposal of integrating different views of conversational behavior, the models to describe them, and the components that are needed for SAL to perceive and act. A prototype implementation of the framework is presented to establish proof-of-concept.

Chapter 2

Modelling head movements

In this chapter a close look is taken at what a movement of the head is, what it looks like, how it can be detected, and how it can be synthesized. Focus in this chapter is thus on the *form* of head movements. As mentioned in the introduction, later in this report more emphasis is put on *functional* aspects of head movements and Listener behavior. Now we will explain first what ‘modelling of head movements’ has to do with SAL.

The Sensitive Artificial Listener (SAL) agent will be equipped with a vision system. This enables it to *see* what is happening and especially what the human Speaker is doing, aside from verbally expressing his/herself. One of the things explored here is how to automatically recognize head movements. A head movement recognition system can be used by SAL, to *perceive* the head movements of the Speaker. The following steps show how this could work:

1. The ‘eyes’ of SAL (a camera) produce a discrete signal (video).

2. For SAL to perceive a head movement, the head movement recognition system needs to attribute a symbolic value to subsequent parts of this signal.

3. The symbolic value from the recognition system now represents the actual head movement and is output to SAL. Based on this information SAL can make a decision on what to do next.

For example, if the Speaker performs a ‘fast nod’, the video registers the head of the Speaker as it makes the nod. The head movement recognition system taking the video as input should then produce the symbolic value ‘fast nod’ for the recognition to be successful. Now, the value ‘fast nod’ is input to SAL so that SAL can decide on e.g. ‘answerring’ to this nod by doing a para-verbal “aha”. In chapter 5, ideas for the implementation of such a decision system for SAL are presented. This system however, is not concerned with synthesizing the movements themselves, but more to produce commands for e.g. a talking head such as RUTH [Douglas DeCarlo et al., 2002] that will responsible for the embodiment of SAL. But which commands should be available to the decision system?

Since the main concern in this chapter is to find out what head movements are like, it will not only cover enabling SAL to *perceive* head movements as they occur in conversations, but also to enable SAL to *synthesize* them. The approach taken here is to equip SAL with a repertoire of head movements for the purpose of perception as well as synthesis. Although Listeners might use head movements differently than Speakers, it is our goal to describe the *form* of head movements from both Listeners and Speakers using the same model.

Furthermore, it is assumed that modelling the *form* of head movements is necessary in analyses linking *form* to *function* in the context of face-to-face conversations, such as described in section 3.4 and chapter 4. This conception is not so much about SAL automatically perceiving and eventually synthesizing head movements, but more about how to accurately capture the movements made during conversations. Therefore, this chapter also deals with annotation schemes of head movements as they are used in other research.

In the first section (2.1) we will look into manual annotations of head movements in a video-taped face-to-face dyadic conversation.The annotations are made on video data from the GT2M database (see appendix A) and describes several behaviors of two human conversants. The transcript is then used to playback the original conversation to get an idea of how well the behaviors were . It is expected that this kind of simulation will give some clues on how to model conversational behavior, specifically head movement behavior.

The next section (2.2) presents a head tracker capable of using video-taped data as input. Head movement data from the SAL database [Roddy Cowie et al., 2005] gathered with the tracker is used to construct a language model of head movements as defined in section 2.3. The construction of the model uses a data collection described in section 2.5, and is further detailed in section 2.4. The model and the tracker are evaluated in section 2.6 with an online survey. The survey contains mimicked head movements from the original videos similar to the simulation presented in section 2.1, but using a different database and *automatically* constructed transcriptions.

2.1 Simulating head movements in a conversation

Figure 2.1 shows a process from raw footage to simulation and analysis as it is partly applied in this section. The data used here is taken from the GT2M data. This data collection contains face-to-face dyadic small-talk conversations. *Data processing* refers to the preparation of the data for annotation and detection. After the signal abstraction process, the abstracted signals can be merged into one data file that can then be used for simulation and analysis. Here the annotation and simulation processes will be described.

The simulation process, as performed here, results in two 3-D head models (RUTH) that imitate the behavior of the participants in the original conversation. Watching this simulation is like watching the actual conversation, only with different performers/actors.

As mentioned in the introduction of this chapter, the purpose of this annotation process is to capture some behaviors during the conversation in a (as much as possible) non-interpretative manner, i.e. focusing on *form*. Which behaviors should be annotated, how, and at which level is to be found out here. To annotate the footage ELAN [Language Archiving Technology, 2007] is used,

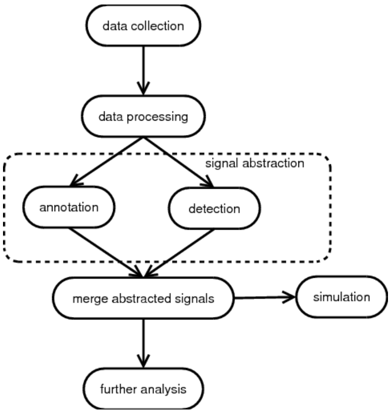


Figure 2.1: process from data collection to simulation and analysis

therefore the terminology and concepts used in this section correspond more or less to the ones used in ELAN.

2.1.1 Controlled vocabularies

The following types of behaviors for both participants are taken into account:

- head movements
- gaze behavior
- blink behavior
- utterances

Of course, this is just a small subset of all behaviors present in the conversations, but it's a start. From this list of behaviors two *controlled vocabularies*, three *linguistic types*, and three *tiers* per participant are devised. The emphasized terms refer to concepts used in ELAN. A controlled vocabulary defines a fixed set of labels that can be assigned to behaviors or events. A linguistic type may use a controlled vocabulary to restrict the annotations made on a certain tier.

In other research, such as e.g. [R. Alex Colburn, 2000], specifying gaze behavior is restricted to the distinction between looking *at* or *away* from a conversational partner. In [Atsushi Fukayama et al., 2002] for example, the *direction* of gaze is also taken into account. In the simulation presented here, direction is also taken into account and a distinction is made as to whether a person is looking *down*, *up*, *left* or *right*, when not looking *at*. During a blink it is difficult to determine gaze direction, so blink behavior is merged with gaze direction behavior. Combinations of up/down with left/right are added, as well as a gaze direction towards the camera. This gives the label set defined in table 2.1.

If one considers the human head as being just a rigid body in three dimensional space, the position of this body can be described by it's *translation* and it's *rotation* along the three axes of this space. If only movements along the

Table 2.1: gaze and blink controlled vocabulary

label	*description*
At	participant is looking at conversational partner
U	up
D	down
L	left
R	right
UL	diagonal up-left
UR	diagonal up-right
DL	diagonal down-left
DR	diagonal down-right
C	participant is looking at the camera

Table 2.2: basic translations and rotations

label	*description*	*label*	*description*
tU	translation up	rU	rotate up (pitch)
tD	translation down	rD	rotate down (pitch)
tL	translation left	rL	rotate left (yaw)
tR	translation right	rR	rotate right (yaw)
F	forward	C	counterclockwise roll
B	backward	J	clockwise roll

three axes are taken, a basic label set for head translations and rotations can be defined as shown in table 2.2.

With the first peeks at the recorded footage and during the first annotation sessions, it became clear that movements are not made along or around just one axis at the same time. Furthermore, it was also noted that RUTH [Douglas DeCarlo et al., 2002] has a built-in set of combination movements, describing a change in position along more axes at the same time, which are based on observed movements. The head movement label set was therefore expanded with the labels listed in table 2.3. These combinations do not cover all possibilities, but the table does contain *observed* movements in part of the data. From the label sets a controlled vocabulary is constructed for use in ELAN.

2.1.2 Linguistic types and tiers

In ELAN a linguistic type may or may not have a controlled vocabulary. A linguistic type can be assigned to a tier. The linguistic type is used to define what kind of annotations belong to the tier that has this type. The following linguistic types were used on the data:

- head movements: uses the head movements controlled vocabulary
- gaze and blink: uses the gaze and blink controlled vocabulary
- utterances: has no controlled vocabulary, the annotator can label the annotation with the uttered words or short sentence.

Table 2.3: combinations of rotations and translations

label	*description*
rLU	rotate left + up
rLD	rotate left + down
tLU	translate left + up
tLD	translate left + down
rRU	rotate right + up
rRD	rotate right + down
tRU	translate right + up
tRD	translate right + down
FrU	forward + rotate up
FrD	forward + rotate down
BrU	backward + rotate up
BrD	backward + rotate down
CrU	roll counterclockwise + rotate up
CrD	roll counterclockwise + rotate down
JrU	roll clockwise + rotate up
JrD	roll clockwise + rotate down

Tiers in ELAN contain the actual annotations that represent events in the footage. A tier has a participant name assigned to it. One annotation in a tier is defined by a start time, an end time and a label complying to the above mentioned definitions of labels and types.

2.1.3 Simulation process

RUTH is able to synthesize several behaviors. Here RUTH is used to synthesize head movements, blinking, lip movement and speech. Initially, no changes were made to RUTH, so the standard "canned animation" facility was used (see the RUTH manual [Doug DeCarlo and Matthew Stone, 2002]). This facility allows for script-like files, with the TMG extensions, to specify on a high-level, which parts of the 3D head model should be deformed and when.

For the simulation process a transformation is made from the ELAN file to a TMG script using an XSLT[1]. The XSLT translates for example the label 'U' into a suitable TMG command for RUTH. The speech is manually translated into phonemes with the help of txt2pho [2], since no XSLT was available to translate words to separate phonemes.

2.1.4 First minute of Lidewij and Rob 2

From the GT2M data the first minute of the second session of Lidewij and Rob's task-oriented conversation is taken, annotated and simulated as described above.

To get a rough idea of what the annotation data looks like, table 2.4 lists the label count and average duration of the annotated behaviors. For Lidewij, head movements, gaze and speech were simulated, for Rob only the head movements. The annotations were translated and animated by RUTH, output to a movie

[1] see XSL Transformations (XSLT) `http://www.w3.org/TR/xslt`

[2] see IKP-Forschung: Phonetik `http://www.ikp.uni-bonn.de/dt/forsch/phonetik/`

Table 2.4: annotation data label count and average duration

	Lidewij		Rob	
	count	duration (s)	count	duration (s)
gaze and blink	67	0.89	55	1.08
head movements	60	0.69	86	0.45
utterances	15	1.73	16	2.35

and then collated as in the original movie. Figure 2.2 depicts this process. The movie is available online as a Flash movie[3].

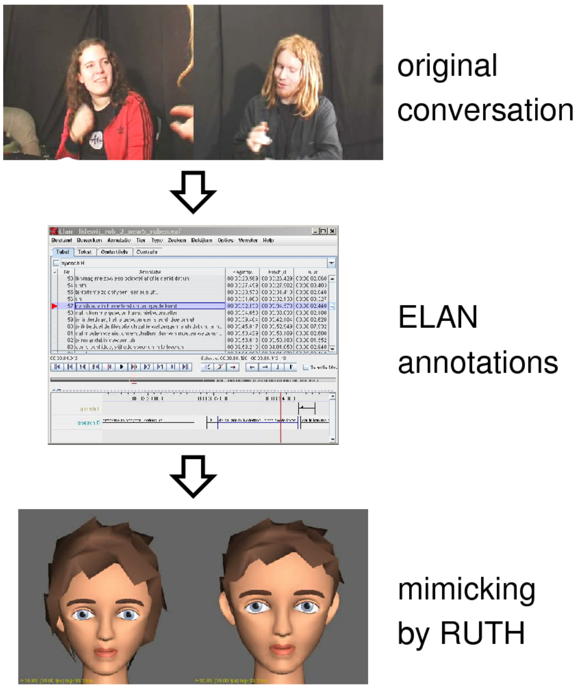


Figure 2.2: from conversation to annotation to simulation

This result was not thoroughly put to the test, but when watching the movie, several people reported an impression of natural interaction. It was noted though that the movements could be more subtle at some times. From this small exercise it was concluded that the label set for head movements, see tables 2.2, **??** and 2.3, *do* capture the direction of the movement, but no information is captured about the duration, size or speed of the movements, hence the lack in subtleness.

[3] see One minute of Lidewij and Rob `http://wwwhome.cs.utwente.nl/~rkooijma/rl_conv.swf`

2.2 A simple head movement tracker

This section reports about a simple head movement tracker developed for the SAL project to capture rigid head rotations. Since the tracker is to be used by SAL in real-time to track movements of a Speaker, it focuses on a simple, speedy and light-weight solution, rather than robustness and accuracy.

In subsections 2.5 and 2.6 the tracker is also used in analyses of head movements in video data. So, as described at the start of this chapter, the tracker is intended to (1) enable SAL to perceive head movements, (2) provide the means for creating a head movement repertoire for synthesizing head movements and (3) aids in analyses of head movements *forms*.

2.2.1 Automatic head movement detection

El Kaliouby [Rana Ayman el Kaliouby, 2005] uses computer vision techniques to automatically recognize facial expressions as well as head movements from a video signal, and then use this for a computational mind reading system. Iwano et al [Yuri Iwano et al., 1996] and Kapoor et al [Ashish Kapoor and Rosalind W. Picard, 2001] show that automatic head movement, or head pose detection is possible with a simple tracking algorithm. The tracker described here focuses on capturing head rotations from video data. It consists of four programs that can be linked together with a pipe, connecting the output of one to the next. The four programs are:

- HMD: plays a movie file and tracks points on the head and body selected by the user
- HEADROT: calculates rotation angles given the position of the tracked points for every frame in the movie
- HEADVAR: filters noise from the rotation angles, and calculates angle velocities based on the last two measurements of the angles.
- HEADFEAT: segments the data from HEADVAR, and extracts some features of the movement in the segment.

How these programs work, is detailed in the appendix, see appendix B. Here, only some basic properties of the programs are discussed.

2.2.2 Head tracker head model

The HMD program allows the user to select points on the head and body by clicking on the movie window, as depicted in figure 2.3. A region around the clicked point in the movie is searched for suitable track points. A total of five regions must be selected by the user:

- left, right eye
- nose, and
- two on the body (e.g. the shoulders)

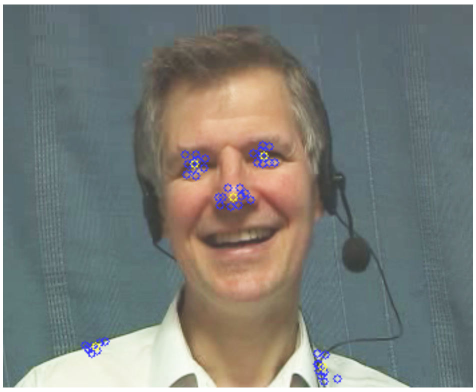


Figure 2.3: screenshot of the point tracking

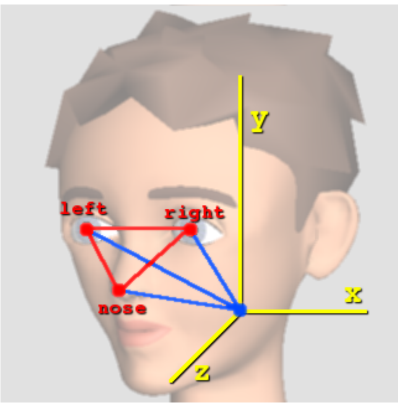


Figure 2.4: tracked points in head model

Figure B.3 shows the head tracker head model. In the model left eye, right eye and nose are connected to a rotation point in the upper neck. This rotation point is considered the origin of a three dimensional Euclidean space. The two points on the body, the 'zero points', act as a reference that determine the position of the origin, and are not shown in the figure. The question now is: how can the coordinates of the points result in rotation angles around the x-axis (pitch), y-axis (yaw) and z-axis (roll)?

2.2.3 Head movement rotation angles

The output of HEADROT is further processed by HEADVAR and HEADFEAT to extract *features* of movement segments. It is assumed that exact rotation angles are not necessary, but an indication will suffice. To give an impression of some tracked movements the following figures (B.7, B.8 and B.9) show some typical movements observed in available video data using HEADROT. The formulas used to calculate the angles as depicted in the figures is detailed in the appendix (see B.3.1).

A nod in the video data (figure B.7) results in rotation angles around the x-axis. This can be seen in the figure as an amplitude change in the rot-x

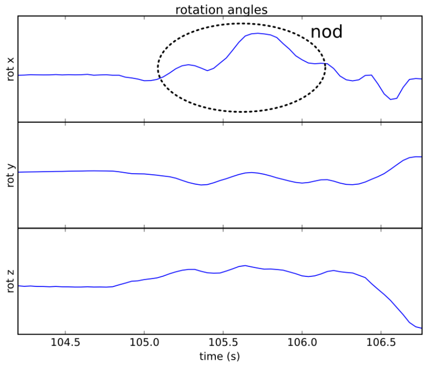


Figure 2.5: nod measured with HEADROT

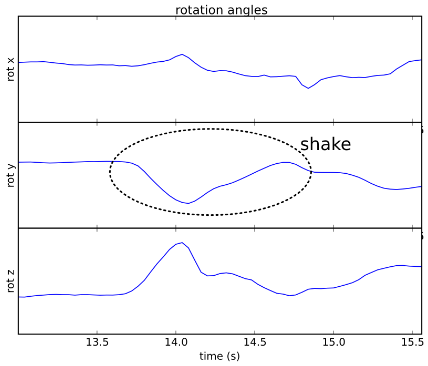


Figure 2.6: shake measured with HEADROT

signal: first the angle gets bigger, meaning an upward movement of the head, then it gets smaller again. Figure B.8 shows the angle indications of a lateral movement of the head. This mainly influences the rot-y amplitude, but as can be seen in the figure, also rot-z is influenced. Using the formulas, particularly for this fragment, the head shake also causes rot-z changes over time because at the start of the fragment the head already had a non-zero rotation angle around the z-axis. This is one of the known shortcomings of the formulas used by HEADROT.

Since the videos are sampled by HMD at a fixed frequency, the HEADROT program outputs the rotation angles at the same fixed frequency. If no change in head position is present in subsequent video frames, the headrot program will output the same rotation angles for each frame. From the observation that no change in position for a certain period in time results in no change in rotation angles, it can be concluded that no movement was present. Since our interest lies in measuring head *movements,* the next program in the chain, HEADVAR, focuses on *change of rotation angles between two consecutive frames.* This change of rotation angles within a fixed period of time will be referred to as the *angle velocity.*

The HEADVAR program works by first smoothing the output of HEADROT to

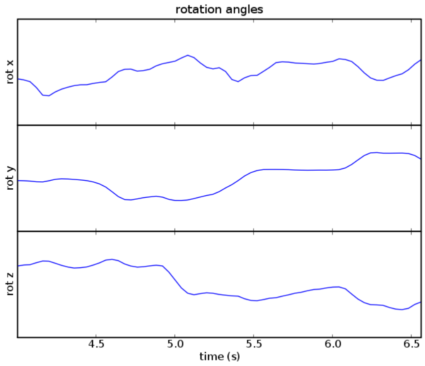


Figure 2.7: complex movement measured with HEADROT

reduce noise using the moving average over a window of three frames. Then the following features are calculated for each frame for each axis:

- angle velocity: change of rotation angles between two consecutive frames, and
- direction

The direction feature has the values -1 or 1 depending on a positive or negative angle velocity. A value of 0 is returned if the angle velocity is considered too small as defined by a threshold. The thresholds can be configured manually for each axis. In the HEADFEAT program the value of this direction feature is used to determine segments in the movement data. Figures B.10, B.11 and B.12 show the same fragments as before, but now the output of HEADVAR is plotted.

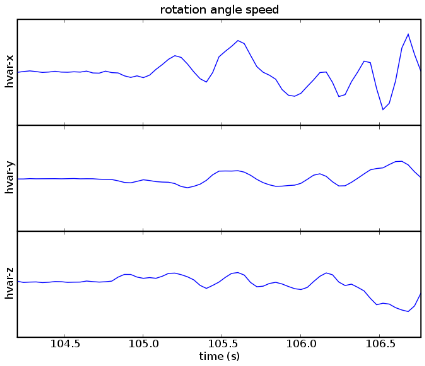


Figure 2.8: nod measured with HEADVAR

2.2.4 Rotation angle velocities to elementary head movements

As stated at the start of this chapter, for SAL to be able to perceive head movements, and for analyses of how head movements are used in conversations, it

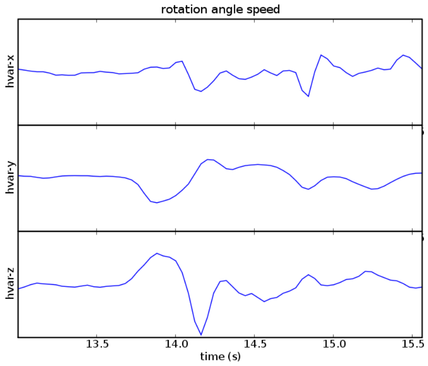


Figure 2.9: shake measured with HEADVAR

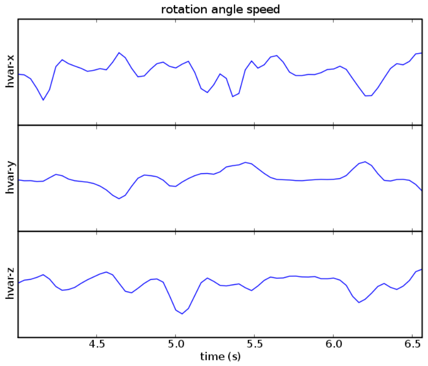


Figure 2.10: complex movement from HEADVAR

is necessary to have some sort of symbolic representation of a head movement. Therefore, the data must be somehow segmented and given this symbolic representation. In section 2 this is done manually using an annotation scheme, while the HEADFEAT program described here, will do this automatically based on the output of the HEADVAR program. But what is this symbolic representation? What will HEADFEAT measure then?

As noted in section 2.1.4 the annotation scheme used there *does* capture the direction of the movement, but is does *not* capture size and speed of the movement. It seems that we need a more fine-grained representation of head movements that incorporates direction, speed and size. For this purpose a new concept is used to describe head movements called elementary head movement (EM).

The HEADFEAT program is developed to measure EMs. Some details about the workings of HEADFEAT are presented in the appendix. To summarize, the program finds EMs in the output of HEADVAR by looking for start and end patterns in the HEADVAR direction values over time. When it has found a segment the following features are calculated:

- *duration* in seconds: as determined by the segmentation,

- *cumulative direction*: sum of HEADVAR directions in the segment,
- *magnitude*: a value at a scale from 1 to 4 based on the average of the absolute angle velocities in each frame in the segment, and
- a *shape* vector describing the progression of the movement along this segment.

The *duration* and *direction* features can be used to get an indication of the *size* of the movement. A movement with a small duration and a small direction value is also small in size, in other words the 'total distance' the head 'travelled' (rotated) is small. The *duration* and *magnitude* are an indication of the *speed* of the movement. A short *duration* with a high *magnitude* means that the movement was fast. The *shape* vector indicates if the movement e.g. starts fast and end slows, or if the movement is very constant.

As an example table B.3 is shown, which represents the HEADFEAT output of the fragment that contains a nod also shown before. Other examples can be found in appendix (see B.4). The nod can be traced to high duration, direction and magnitude values in the x columns. The bold faced rows in the x columns (t=105.84 and t=106.16) show first a positive direction attribute (*dir*) followed by a negative, furthermore the magnitude column (*mag*) has value 4. This means the head first moved upward then downward and that this was a big movement, meaning high amplitude. However, around the same time, the y columns also show a similar pattern while in figure B.8 the amplitude of rot-y is not that high. Probably this is due to incorrect threshold values used by:

1. HEADFEAT for determining the magnitude category and
2. HEADVAR for determining the direction value.

Since these thresholds were chosen arbitrarily, the output of HEADFEAT may not be sensitive enough to distinguish between nod or shake. If for example the threshold for the x-rotation is too small, but the threshold for the y-rotation is too high, and assuming that a movement is made along the x-axis (a shake), the headfeat program might also see y-rotations because of noise, which could then be translated to nod. So, because the output of HEADROT and HEADFEAT are not yet checked against *ground truth* data on head movements, it is still unclear how well they perform.

2.2.5 Using the head tracker chain

This section has shown a selection of four programs developed to be used in projects that need to measure and analyze head movements. The programs can be chained together with a *pipe* to work on video footage and produce data about the head movements made.

Furthermore, the concept of an elementary movement (EM) presented here, allows for both automatic recognition and enriched analyses of head movements. By taking speed and size into account, an EM is still more abstract than sequences of rotation angles, but also more precise than using terms like *nod* or *shake*.

The manually set threshold values used in HEADFEAT and HEADVAR may need adaption to give satisfactory results, this will be explored in the next

Table 2.5: segment containing nod from HEADFEAT

	x						*y*						*z*					
time	*dur*	*dir*	*mag*	*shape*			*dur*	*dir*	*mag*	*shape*			*dur*	*dir*	*mag*	*shape*		
105.48	0	0	0	0	0	0	0.32	8	4	0.49	0.75	0.69	0	0	0	0	0	0
105.52	**0.08**	**-2**	**1**	**1**	**1**	**0.6**	0	0	0	0	0	0	0	0	0	0	0	0
105.56	0	0	0	0	0	0	0	0	0	0	0	0	0.12	-3	1	0.74	1	0.74
105.76	0	0	0	0	0	0	0	0	0	0	0	0	0.16	4	2	0.77	0.85	0.56
105.8	0	0	0	0	0	0	0.28	-7	4	0.6	0.83	0.57	0	0	0	0	0	0
105.84	**0.28**	**7**	**4**	**0.65**	**0.86**	**0.57**	0	0	0	0	0	0	0.08	-2	1	1	1	0.47
106.12	0	0	0	0	0	0	0.28	7	4	0.59	0.81	0.59	0	0	0	0	0	0
106.16	**0.32**	**-8**	**4**	**0.63**	**0.85**	**0.57**	0	0	0	0	0	0	0.16	-4	2	0.77	0.81	0.53

subsections. Also some other problems with tracking the movements might arise. We argue though that, with adapted threshold values, the presented programs could provide useful input for offline as well as on-line (real-time) analyses of head movements in conversations.

2.3 Head movement features

While chapters 3 and 4 deal with the question of what to do when a head movement is detected, they don't research what makes up a head movement, and how head movements can be detected.

To summarize, in section 3.4 of this report head movements are described using the following features:

- type: nod, shake, move, sweep, roll, waggle
- direction: up, down, side, repeated
- shape: fast, slow, jerk, big, arch

In section 4.2 a head movement is described by a rotation axis (x,y,z) and a shape (nod, move, nod with overshoot). And, as we have already seen in the previous section (2.1), yet another annotation scheme describes head movements. This time by basic rotations and translations along the axes of three dimensional space, and combinations thereof (see tables 2.2, **??** and 2.3).

However, based on a simulation of a conversation using this last annotation scheme it was concluded that the scheme lacked subtleness to accurately describe which head movements were made. It is not unthinkable that this might also holds for the other schemes used in this report. Therefore, in the remainder of this chapter an effort is made to gain more insight into subtleties of head movements using the concept of elementary head movements (EMs), as introduced in the previous section. The EM measurements from the head tracker are tested for automatic recognition and synthesis to be used by SAL, as well as, user perception of synthesized movements.

2.3.1 Model requirements

Here it is first explored in more detail how the raw signal from the HEADVAR program can be modeled into an abstraction of the signal. As mentioned at the start of this chapter, abstraction is necessary because, for SAL to be able to perceive what's going on, it needs to consider not just one sample of rotation angles, but a sequence of rotation angles (*token*). The token can be used in a recognition module to ascribe some known head movement *type* to it based on the features present in the *token*. Here some requirements are presented for the modelling of head movements in terms head movement types (abstraction) and tokens (original). The requirements for the model are:

1. it allows for the original signal to be reproduced using the abstracted signal,
2. the abstracted signal can easily be translated into commands for a talking head (in our case RUTH),

3. the abstracted signal is rich enough to be able to reproduce the original movement in such a way that the loss of subtleties in the perception of the movement is acceptable.

Requirement 2 aims for an abstraction of the original signal such that high-level commands can be generated to be used on a talking head. In this way, a collection of tokens can be gathered as a basis for producing a head movement repertoire. For example, RUTH has a high-level command called *jog* and, because this is a *high-level* command, it only requires a few parameters to specify what kind of movement should be made. The question then is what the values of these parameters should be to produce *natural* head movements. So, if we have a database of head movements used by real humans, RUTH could use this to produce *natural* head movements.

While requirement 2 drives the model towards having just a few parameters to produce head movements, requirement 3 on the other hand dictates that this simplification cannot go on too far. In other words the original movement cannot be corrupted by the abstraction too much. This last requirement is quite a subjective criterion, nonetheless in section 2.6 an effort is made to get a grip on how the head tracker with the model performs in terms of the requirements stated above and how it could be improved.

2.3.2 Head movement language model

Looking back at figure 2.5 it can be seen that a label 'nod' (type) is given to the signal (token). The figure was generated from a fragment of the output of the HEADROT program. What happened here was that (1) the video was searched for a nod, and (2) the fragment containing the nod was manually selected from the HEADROT output and plotted. So first we defined a label (nod), and then part of the signal that could be given this label was selected.

Attributing labels (classes) to a signal is a typical *classification problem.* However, one important issue is that there is no consensus on a definite set of classes describing head movements; a head movement typology. One could argue that it compares to the problem of speech recognition without being able to say which words should be recognized, because no words are defined yet. For example: if Alice and Bob are communicating and Alice says P, for the communication to be successful Bob has to perceive what was being said as P and not e.g. Q.

So which P's and Q's are we talking about? Here it is suggested that to answer this question the signal needs to be (1) segmented, (2) features must be extracted from the segment and (3) segments could be combined to form *meaningful units.* The head movement language model proposed here is constructed by the following processes:

1. segmentation: determines the start and end point of a head movement,

2. feature extraction: calculates the relevant features present in the raw signal,

3. matching tokens to types: compares a perceived segment with known head movement types,

4. combining recognized types: estimates the most probable sequence of recognized EMs based on a head movement language model.

The first two processes are briefly touched upon in section 2.2.4. Processes 3 and 4 are detailed in the next section. In section 2.6, these processes will be automatically performed on data from the HEADVAR program (see 2.2.3) and the recognized movements will be evaluated. The data collection is explained in section 2.5. Now the implementation of the above mentioned processes will be described shortly.

2.4 Implementation

To be able to test the head movement language model, a few Python programs are written that automatically perform the processes mentioned above. One of the executables is capable of constructing a language model based on found elementary head movements in some training data (MAKEHEADLM) and another takes this language model and observations from HEADVAR to produce the most likely sequence of EMs (MOVETOMOVE). The language model consists of a unigram model that contains counts and estimated probabilities based on a training set.

In this initial implementation, calculating the most likely sequence of elementary head movements does not include using a more sophisticated model that also takes bigrams, trigrams, or some grammar based rules into account.

2.4.1 Segmentation and feature extraction: headfeat

Segmentation and feature extraction is performed by the HEADFEAT program which is shortly explained in chapter B.4. The segmentation works in two steps by:

1. setting a minimum threshold on the amplitude of the HEADVAR signal: the HEADFEAT *direction* feature.

2. Finding patterns in the stream of HEADFEAT *direction* values. Start and end patterns can are defined in HEADFEAT by using regular expressions.

The thresholding is is depicted in figure2.11. Using a threshold on some data results in a sequence of 0, -1 or 1s. From the figure, the signal on the x-axis (hvar-x) produces something like the following sequence:
000000000011000111000-1-1-100-1-1000-1-1-1011.

The pattern matching works on a window of size 3, 4 or 5 taken from the HEADVAR sequence. Then the windows are matched against the start and end regular expressions. The regular expressions are defined such that a change of movement direction results in an end of the previous segment, and start of a new one. If no head movement is present an end is detected by a sequence of 0s, a new start is then detected by a sequence of 1s or -1s.

From the segment headfeat calculated the head movement features. The following features are used here:

- duration (*dur*)

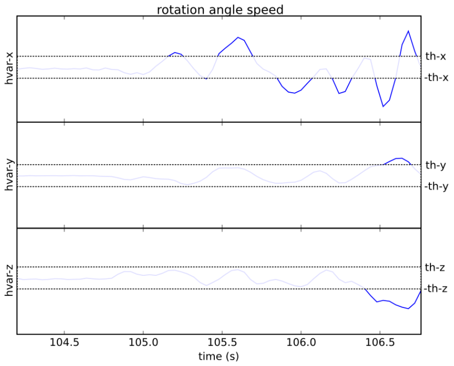


Figure 2.11: thresholds on the HEADVAR signal (nod)

- direction (*dir*)
- magnitude (*mag*)

In this implementation a segment is presented as a vector. Which is built up as follows:

$\begin{bmatrix} time & dur_x & dir_x & mag_x & dur_y & dir_y & mag_y & dur_z & dir_z & mag_z \end{bmatrix}$

An example of multiple vectors from the HEADFEAT program that also include a shape description can be found in tables B.3.

2.4.2 Constructing an EM dictionary: makeheadlm

An EM dictionary refers here to the set of head movements *types* a head movement *token* can be matched against as explained in section 2.3.1. Here, the EM dictionary consists of nothing more than a list of earlier observed EMs with their occurrence counts and a priori observation likelihood estimations. The list is constructed using the MAKEHEADLM program. This program takes as input a training set of HEADFEAT samples, and outputs a unigram model. The model is constructed by comparing HEADFEAT vectors with each other. The comparison works as follows

1. use the *score* function on each pair of observation vectors (from HEADFEAT)
2. if the score (see below) is higher than a given threshold, add one to the count of the first vector and store the score in a data structure (score map), mapping one observation to a list of similar observations
3. filter out vectors that have lower score compared to other vectors in the score map, such that no vectors are left that have a low score in the score map.
4. estimate the observation likelihood of all left-over vectors (see below).

The *score* function:

$1 - \frac{|em_1 - em_2|}{|em_1| + |em_2| + 1}$

returns a value between 0 and 1 that gives an indication of how much vector em_1 and em_2 are alike. The *time* element of the vector is set to 0 before comparison. $|v|$ calculates the sum of the absolute values of the vector v (the L1-Norm[4]). An outcome nearing 1 means the two vectors are similar, 0 means they are not similar at all. The score function has the following properties:

- If the difference of each of the elements from $em_1$and em_2 is small, the numerator of the division is also small; the fraction will then be close to 0. If the difference of each of the elements is big the numerator will also be big.
- The denominator adds the absolute values of the vector elements. If the elements are big the denominator will be big, resulting in a fraction closer to zero than with smaller values for the elements. This compensates for the numerator being bigger if the difference is caused by large element values.

Normalizing the vectors and calculating the Euclidean distance function could also be used here. This however would be computationally more complex which is not preferable since SAL must be able to response to the movements in real-time, as quickly as possible if needed.

The a priori observation likelihood estimation is calculated by $\frac{em\ count}{total\ count}$, where *em count* is the count of similar vectors in the training set, and *total count* the total number of vectors. This value will be referred to as p in the remainder of this chapter. No smoothing or back-off is used to compensate for unobserved EMs.

2.4.3 Matching tokens to types: movetomove

The MOVETOMOVE program takes a unigram model and observed head movements (*tokens*) as input, and outputs a sequence of elementary head movements (*types*), and a translation of this sequence into the TMG format used by RUTH [Douglas DeCarlo et al., 2002]. The TMG format is a list of commands that are translated into deformations of vertices over time of a 3D head model.

The observed head movements are translated into EM vectors by HEADFEAT. Then for each observed EM the unigram model is searched for the best match. This means all uni-grams in the unigram model are compared with the observed EM using the *score* function. The unigram giving the highest score is chosen as the best translation for the observed EM.

If we want to reconstruct the original signal, the time between subsequent EMs also has to be taken into account. After the segmentation process, the point in time a segment is detected is known, as well as its duration. This information can then be used to reconstruct the original signal.

As stated in 2.3.1, one of the requirements is that the abstracted signal can be used to produce high-level commands for a 3D talking head. In our case we use RUTH for generating head movements. RUTH has a command called *jog* that instructs the head to perform a move. The RUTH manual [Doug DeCarlo and Matthew Stone, 2002] has more details on this command. Important to note here is that it takes the following parameters:

[4] e.g. see for quick reference L1-Norm `http://mathworld.wolfram.com/L1-Norm.html`

- tstart, tend: gives the start time, and implies the duration
- attack, decay, f: shape of the movement
- x, y, z, d: quaternion rotation vector for the direction of the head rotation movement, where x, y, z give the axis of rotation and d the angle in degrees.

Figure 2.12 is taken from the RUTH manual, and depicts how the parameters are used to produce a transient motion. A transient motion is a movement that starts end ends at a neutral position. So now the features of the EM vectors need to be translated into parameters for the *jog* command. The MOVETOTMG program does this by treating each axis separately, thus leaving two of the x, y, z parameters 0. The d parameter is a function of the magnitude EM parameter. The f parameter is a constant. And tstart, tend, attack and decay are all a function of the duration. The output of the program is a sequence of TMG commands (only *jog* for now), which can be used on RUTH.

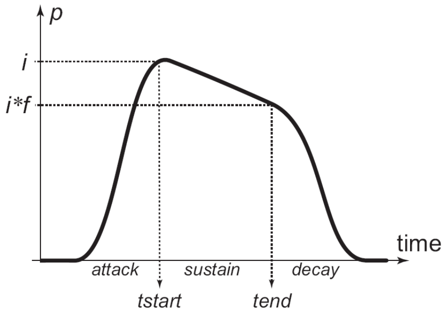


Figure 2.12: RUTH transient motion

2.5 Data collection

Since the implementation is to a large degree dependent on the correctness and accuracy of the HEADROT and HEADVAR programs, it is unknown how well they perform. This, together with the manually set threshold parameters for the segmentation and feature extraction process, makes it quite questionable if this model will perform well. Here, some data is collected and the MAKEHEADLM program is used to construct a first language model of head movements. In section 2.6 a small evaluation of this model is presented.

2.5.1 First unigram model

The data used in this experiment is taken from the SAL database [Roddy Cowie et al., 2005]. From the database, the first five minutes of three sessions were selected from three different persons chatting with the SAL character Prudence. Table 2.6 lists the total amount of video frames analyzed by the segmentation algorithm and the amount of segments detected. Using the MAKEHEADLM program a unigram model was constructed as explained above (2.4.2). A total of 240 uni-grams were found, 60 of which have a count greater than or equal to

Table 2.6: segments from the segmentation process

session	*time*	*measured frames*	*EM segments*
all	15m	22312	5573
Roddy	4m40	6980	1670
Ed	5m20	7429	2007
Ellen	5m	7903	1896

10.To give another impression of the contents of the model table 2.7 lists the top 10 based on the *p* values. The format of *type* is:

Table 2.7: unigram top 10

	type	*p*	*count*
1	<[X{2,2,1}]>	0.0389	217
2	<[X{5,4,1}]>	0.0330	184
3	<[X{3,3,2}]>	0.0309	172
4	<[X{2,-2,1}]>	0.0301	168
5	<[Z{2,-2,1}]>	0.0294	164
6	<[Z{2,2,2}]>	0.0275	153
7	<[X{7,6,2}]>	0.0269	150
8	<[Z{2,1,1}]>	0.0251	140
9	<[X{4,4,4}]>	0.0251	140
10	<[Z{1,-1,1}]>	0.0239	133

<[axis{duration,direction,magnitude}]>

As can be seen in the table, movements around the x-axis are numerous, while movements around the y-axis are not even in the top 10. Most movements in the top 10 are movements with a small magnitude and not too long duration.

2.6 Evaluation

To evaluate if the head tracker with the unigram model presented above are capable of mimicking real head movements from a video, a short survey will be performed. By mimicking head movements performed by a human, it is assumed that we may get some idea if:

1. the correct rotation angles were measured by HEADROT,
2. useful thresholds are used for HEADVAR, HEADFEAT and MAKEHEADLM,
3. the MOVETOMOVE program finds good matches between the observed movement and the elementary movements present in the unigram model, and
4. if the translation of EMs to RUTH *jog* commands and the synthesis of movements by RUTH is perceived as a good imitation of the original movement.

This is probably too much to verify in just one experiment and get reliable results at the same time. Nonetheless, it is assumed here that by asking for some user feedback in a short survey, it would still be feasible to at least get an idea of the performance of the head tracker and the unigram model, and which parts need improvement. Also it would be interesting to see which features of head movements the respondents as sensitive to. In other words, will humans observing head movements notice differences between a human or an artificial performer, and which difference are noticed most?

To answer these questions, participants of the survey were asked for their opinion on the performance by presenting them with fragments of the original movie (from the SAL database) and a mimicked version of the movements produced by the MOVETOMOVE, MOVETOTMG, RUTH chain.

2.6.1 Online survey setup

The unigram model presented in 2.5 is used as the language model. The fragments were selected from the same data as the training set for the unigram. Each person (Roddy, Ellen and Ed) is in 5 fragments. The 15 fragments have an approximate duration of 2 seconds. The respondents were asked for their opinion on each of the 15 fragments.

Beforehand, the group of respondents is divided in two: (1) students and employees from the HMI department, most of which have experience with ECAs, and (2) my colleagues (internet technology specialists) and family. The distinction is made because in reports such as [Z. Ruttkay et al., 2002] Ruttkay et al. argue that people having no experience with ECAs are likely to respond different than people that do have experience with ECAs.

With each fragment the following two questions were asked:

1. How well do you think RUTH imitates the movement?

2. What should be improved about the imitation?

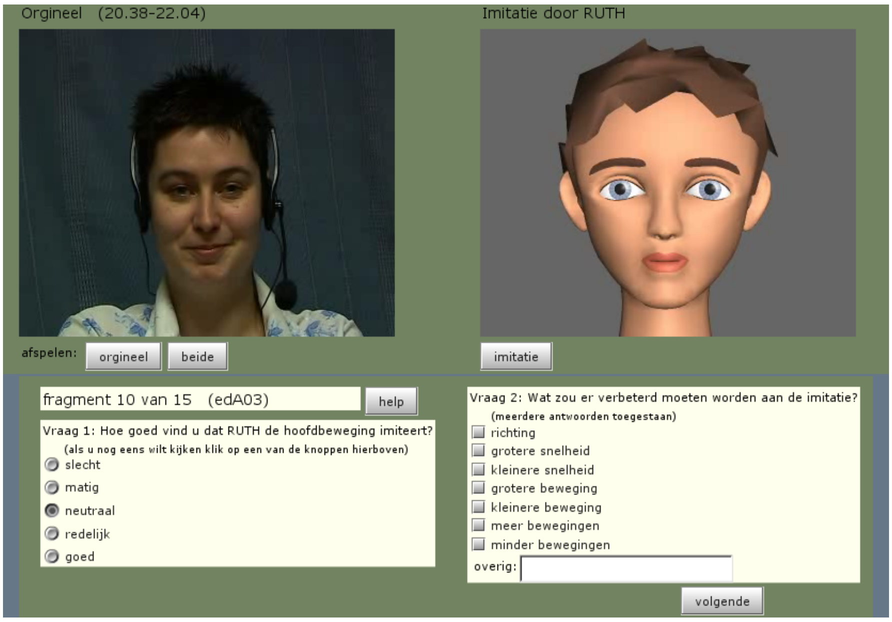
A screenshot of the on-line evaluation tool is depicted in figure 2.13. It is expected that question 1 will give some overall score of how well the tracker and the model perform. Question 2 will be important to get insight in the current problems with the tracker and model, and how people perceive them. In question two, users can select zero or more of the following features of the movement that need improvement:

- direction

- amplitude

- speed

The respondents can also indicate if they find movements missing in the imitation, or if RUTH produces movements not present in the original. An extra text field is added for other problems users might find.



fragment 10 van 15 (edA03)

Vraag 1: Hoe goed vind u dat RUTH de hoofdbeweging imiteert?
(als u nog eens wilt kijken klik op een van de knoppen hierboven)

○ slecht
○ matig
● neutraal
○ redelijk
○ goed

Vraag 2: Wat zou er verbeterd moeten worden aan de imitatie?
(meerdere antwoorden toegestaan)

□ richting
□ grotere snelheid
□ kleinere snelheid
□ grotere beweging
□ kleinere beweging
□ meer bewegingen
□ minder bewegingen

overig:

Figure 2.13: screenshot of the evaluation tool

2.6.2 Survey results

The extra text field was used by eight out of fourteen respondents. The following list summarizes the comments given:

1. While the original fragment had barely any head movements, the movements made by RUTH are too big and too wild.
2. RUTH also needs to move her chin, mouth, body/torso, lips, eyes, eyebrows, and it should blink.
3. Movements need to be more subtle.
4. After the original stopped, RUTH keeps moving for a while.
5. The start of the movement is completely different from the original
6. The timing of the movements is not good.
7. RUTH seems to move when the eyes are blinking.

From some of these comments it can be concluded that not all respondents were looking *only* at head movements. This could mean that the assignment was not always clear, or the absence of other behaviors was distracting. It could also mean that attention *was* paid to the head movements, but the respondent was just trying to be helpful by giving extra comments.

The overall average score for question one on a scale of 1 (bad) to 5 (good) was 2.89, for the first group (researchers and students) this was 2.79, for the second (colleagues and family) it was 2.97. When grouping the results by fragment, it can be seen that some fragments clearly score higher than others. This is also

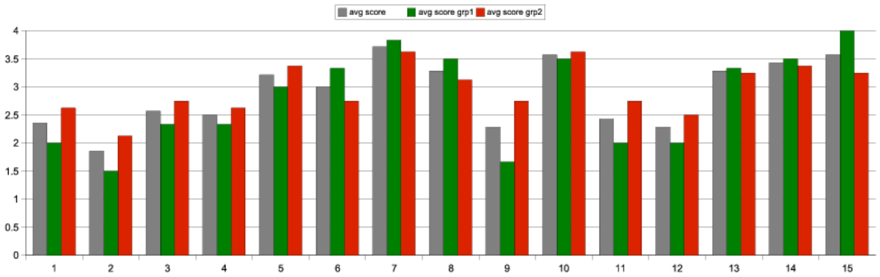


Figure 2.14: average scores per fragment on question 1

depicted in figure 2.14. Fragment 2 scores worst with 1.86, while fragment 7 scores best with 3.71.

Figure 2.15 gives a visual representation of the results for question two per fragment. Here the score on the y-axis represents what percentage of respondents selected the feature as a should-be-improved feature.

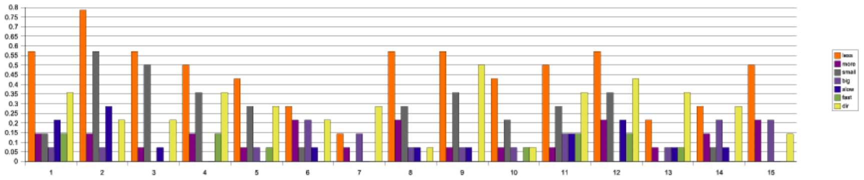


Figure 2.15: average scores per fragment on question 2

It can be seen that the ‘less’ feature was selected the most in almost all fragments: in thirteen out of fifteen fragments this feature was selected most. This means the respondents agree on that, in general, RUTH should perform less movements to better mimic the original fragment. Looking back at the comments listed above, we can link comments number 1, 4 and 7 to this problem. To elaborate, it seems RUTH has problems when (a) there is not much movement present in the original, (b) the movement generation (MOVETOMOVE) takes too many movements into account, especially at the end of a fragment, and (c) eye blinking disturbs accurate head rotation detection by the head tracker programs (HMD and HEADROT).

Other features that score high (meaning bad imitation) in the survey are ‘direction’ and ‘small’. Both may be related to the ‘less’ feature. Additional problems noted with the fragments having high scores on these features are (d) RUTH already starts out with wrong orientation of the head (see also comments 2 and 5) due to MOVETOMOVE assuming a neutral position at the start and (e) subtleties are not well captured by the tracker or the model.

Another way of looking at the results from the survey is to see if RUTH mimics one person better than another according to the respondents. Table 2.8 lists the scores for question 1, figure 2.16 for question two. Most notable is that Ed scores highest on question one, meaning good mimicking, and that Ellen scores worst, especially on the ‘less’ and ‘small’ features. The biggest difference between these two persons that comes to mind when looking back at the

Table 2.8: question 1 scores per person

person	score
Ellen	2.46
Roddy	2.86
Ed	3.36

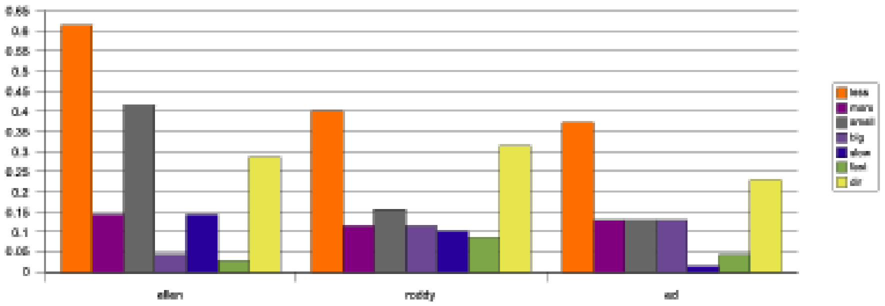


Figure 2.16: question 2 scores per person

original fragments is that Ellen anyway performs less and smaller movements, and that Ed uses head movements much more than Ellen. As the tracker and model, configured as they were, seem to perform bad on subtle movements, this observation also explains why the mimicked version of Ellen, almost keeping her head still, is worse than that of Ed, performing big head movements most of the time.

2.7 Conclusions

This chapter was about *head movement form.* Manual annotations of head movements were made using an annotation scheme that covered translations and rotations of the head in several directions. From a simulation of one minute of a GT2M session based on these annotations, it was concluded that although the scheme is quite extensive in it's descriptions of head movements, it lacks subtleness observed in real conversations, especially head movement features such as size, speed and duration.

Next, a head tracker was described which was used to estimate the rotation angle velocities of head movements. The tracker is capable of estimating rotations along three axes from a monocular video source. Estimations/measurements are output at the same rate as the video frame rate. To be able to make a fine-grained representation of head movements, based on output from the tracker, the notion of ELEMENTARY HEAD MOVEMENT (EM) was introduced. An EM describes a head movement by the following features: duration, direction, magnitude and shape along three rotation axes.

The HEADFEAT program segments the head tracker output and calculates the EM features in real time. This program could be used by an ECA such as SAL to *perceive* head movements. Another purpose for HEADFEAT is the construction of a head movement repertoire for the *synthesis* of head movements. Using videos from the SAL database a unigram language model of head movements was constructed.

Next, the MOVETOMOVE program was used to match *tokens* from the videos of the SAL database against *types* present in the head movement unigram model. The recognition of these head movement types (observed EMs) was output to the 3D talking RUTH to enable an evaluation of the HEADTRACKER, HEADFEAT and unigram model. The results from an online survey with fourteen respondents showed that the head tracker and HEADFEAT program need improvements mainly on: robustness and accuracy. The biggest problems were: detection of nods when eye-blinking occurred due to the fact that the head tracker uses the eyes as a reference for calculating the head movement rotations, too big movements when the original video showed only small movements, and wrong orientation offset in the segmentation process (always neutral) when people had a non-neutral head orientation at the start of the segment.

Despite the poor results from the automatic head movement recognition programs, it is believed here that the notion of elementary head movement is a valuable one, because it captures more features about head movements than various annotation schemes. More accurate and robust tracking and segmentation algorithms will make it possible to automatically measure EMs, which could be valuable for use in a Sensitive Artificial Listener by providing it with the means to *see* how a human Speaker moves his/her head, and for the construction of a head movement repertoire.

Chapter 3

Interpretation & classification of speaker behavior

The goal with this thesis is to model the behavior of a Listener in conversation with a Speaker, but because having a conversation is a social activity, the Listener has to be able to perceive the Speaker and take his/her actions into account. Taking the actions of the conversational partner into account means that decisions on how to react to these behaviors are based on interpretations of the meaning of these behaviors in the context of the conversation. The main question to be answered in this chapter is: which Speaker behaviors are present in a face-to-face conversation, and what do they mean; how can these behaviors be classified and interpreted?

Focus here is on a *framework* of classification and interpretation of *head movements* of Speakers, during face-to-face conversation, incorporating different views on conversational behavior. The framework proposed here can be applied to analyses of conversational behavior in which incorporating multiple views of functions of these behavior is a key issue. It is assumed that models of Listener behavior to be used in the SAL project are aided by a rich account of different meanings that can be ascribed to Speaker behaviors. To demonstrate the framework's use in a functional analysis, it is applied to some data from the SAL database. Then the classifications resulting from this *functional* analysis are linked with *forms* of head movement behaviors of the Speaker. Insights about Speaker behavior gained from the analysis can be used in defining models of Listener behavior, especially for automatic interpretation of head movement forms.

First in section 3.1, it is shortly explained what classification and interpretation of Speaker behavior has to do with the modelling and design of a SAL agent.

Then in section 3.2, it is explained what exactly is meant here by a *framework of classification and interpretation.* Also the motivations for specifying the framework and links with other research are touched upon.

Section 3.3 is about interpretation of behaviors. This section points out how 'reading the mind' of a Speaker, interpreting his/her intentions, is to some extend possible, and what the general implications are of such a practice.

In the last section (3.4) an analysis of some video data from the SAL database is performed on aspects of cognition, social determinants, linguistics and emotion. Also it will show how head movement forms can be linked to the functions they have in the analyzed conversations.

3.1 ECAs listening to Speakers

This section will elaborate on what classification and interpretation of Speaker behavior has to do with the design of ECAs in general, and more specific with the design of a Sensitive Artificial Listener. Later in this chapter, in section 3.4, different aspects of Speaker head movements will be highlighted.

3.1.1 Sensitivity

Designing an Embodied Conversational Agent capable of acting as if it were a human Listener poses some interesting questions like: What does a human Listener do? How should SAL do this?

For many years already people have been drawn to the idea of creating machines that mi mick human behavior. But, although technological developments enable us more and more to make use of the power of automated computation, systems that use this power still have to be programmed by humans. So the answer to the question of what SAL should be capable of doing is not solved just by adding some hardware; the characteristics of the software to be programmed is all important.

In [Dirk Heylen, 2005b], Heylen lists some recent efforts on incorporating human characteristics into ECAs and notes that:

> Increasingly we have come to view language as social action. Behaviours of agents are not only designed for their communicative functions (providing information on the task, regulating conversational flow) but the conversation is part of a social encounter.

The following list sums up some of the characteristics related to conversation as a social encounter. Heylen mentions: engagement, impression, emotion, (social) believability, rapport, friendship, dominance, power, face, et cetera. In this work of Heylen as well as in this thesis it is assumed that getting grip on these characteristics, will provide researchers designing models for ECAs the tools to make ECAs more socially aware. In this thesis specifically it will put some sensitivity in SAL.

3.1.2 SAL as a model-based agent

As the above suggests, ECAs need to incorporate human-like characteristics to be perceived by humans as a person. Blaise Pascal, surely not in the context of designing ECAs, once said:

> When we see a natural style, we are astonished and charmed; for we expected to see an author, and we find a person.

But alas, for SAL to be perceived as a person, it still needs to have an author and be defined by a computational model.

As a basis for modelling SAL, a model-based agent architecture [Russell and Norvig, 2003, p.48-49] is used in this thesis. Later in this report, in section 5.1, an implementation of such a model-based agent is presented. For now, it is important to note that we assume the agent works by performing the following processes:

1. The agent has to be able to use sensor input, such as a camera, to be able to perceive what's going on in the environment. This produces information about the environment and specifically about the behavior of the conversational partner. Terms used here: OBSERVATION, PERCEPTION, CLASSIFICATION.

2. The available information can be applied to a set of rules to produce new information, or change the internal state of the agent. Terms used here: CLASSIFICATION, INTERPRETATION, INFERENCE

3. The available information, including the internal state, can be applied to a set of rules to produce *actions*. Terms used here: DECISION, ACTION, GENERATION

The remainder of this chapter is more about how to *interpret* movements of the head when performing analyses of face-to-face conversations *as a researcher*. One could argue however, that the principles for performing this analysis, and the knowledge gained from it will also apply to SAL and how it could do this automatically. To illustrate: systematically analyzing behaviors as they occur in conversations, a researcher may (a) observe the behaviors, (b) interpret and classify them by labeling the behaviors, and (c) further analyze the behaviors to come up with a model explaining them. SAL, as a participant in the conversation, is not just randomly choosing some actions, or running a pre-defined sequence of actions, instead it will do something similar to the what the researcher does. It (a) observes the conversation, then (b) classifies and interprets the actions of the Speaker (c) puts it in a model describing the environment (designed by the author of the model), to eventually decide what action to perform.

If we can come up with a sensible, structured way of classifying and interpreting Speaker behavior, we could automate this process thus enabling SAL to really see what's going on. One of the things explored in this chapter (section **??**) is the link between head movement forms and head movement function. The knowledge gained from this exploration could be used by SAL to make some inferences about the Speaker's state of mind, including such things as the Speaker's view on how SAL is perceived by the Speaker.

To give an example, consider the following: While the Speaker is talking, he/she may want to check if SAL still understands what's going on. The Speaker does this by gazing and moving his/her head towards SAL. SAL observes this head movement and uses the knowledge about this particular kind of head movement *form* to infer that the Speaker wants to know if SAL is still listening.

3.2 Modelling functions of behaviors

In the Sensitive Artificial Listener project [Roddy Cowie et al., 2005] and in the HUMAINE network [Marc Schröder and Roddy Cowie, 2005] ideas were put forward to model listener behavior and to be able to generate it. In the project, research interests focus on expression and manipulation of emotional behavior. Here, expressing emotion is considered one aspect of behavior among many employed in conversations; this chapter takes a broad view on these behaviors, especially head movement behavior of the Speaker. While the previous chapter (2) focused on the form of head movements, this and the next chapter will focus more on function; i.e. what the purpose of specific head movements could be, what they could mean, what and how they contribute to conversation, and in what context they are used.

3.2.1 Specifying a framework of functions of behavior

The idea for specifying a framework in which the behavior itself (the form) can be linked to it's function is not new. In [Hannes Vilhjálmsson and Stacy C. Marsella, 2005] for example, a proposal is made for a Social Performance Framework which uses the Functional Markup Language (FML) and the Behavioral Markup Language (BML). Vilhjámsson and Marsella argue that requirements of manageability of complexity and flexibility for the design of ECAs is aided by a clear distinction between functional and behavioral aspects of communication. By defining clear interfaces between modules of ECA systems, different architectures, possibly focusing on different aspects of behavior, can be integrated thus contributing to 'overall social believability'. The framework presented in this chapter also addresses this by the distinction between form and function. On the other hand this chapter will be less concerned with the *generation* of Listener behavior and focus more on structured analyses and automatic interpretation of behaviors of the Speaker.

In [Dirk Heylen, 2005b] and [Heylen, 2006], theories about behavior during conversations, head movements and gaze specifically, are viewed at from the point of conversation and language as social action, and how functions of behaviors can be structured by general principles. Heylen summarizes some conventions and rules from different research traditions and domains that are defined to explain how conversations are organized. The findings from all these disciplines should be taken into account, in his words [Dirk Heylen, 2005b]:

> An important challenge for the research on embodied conversational agents is how to integrate these ideas, observations, and theories from the various disciplines and how to put them into rules and procedures that embodied agents can use in actual interaction.

Other inspirations of how to interpret speaker behavior in the context of not only describing behavior, but also *defining* a computational model, comes from the work of Kristinn R. Thórisson [2002] and Christopher Peters et al. [2005b]. The computational models of behaviors used in this research are based on findings in sociology, (social) psychology and linguistics. However, in our opinion, the behavior descriptions still focus on a *limited amount of aspects*, thereby neglecting the diversity and richness of behaviors used in real human conversation. In contrast, studies such as [Evelyn Z. McClave, 2000] and [Nicole Chovil, 1991] focus more on *descriptions* of behaviors and their functions. Models defined in these works are quite *extensive* and directly based on *observations*. This kind of descriptions however, are less suited for use in artificial characters such as SAL because of the restriction of computability. The intention here is to fit the different aspects together and to be able to define a *computational* model for classification and interpretation, but also to not neglect the complexity conversational behavior naturally brings about.

3.2.2 Determinants

To research and describe what interactional behaviors are present in communicative settings, many researchers use a scheme to ascribe functions to behaviors in a systematic way. Some efforts of designing and applying such scheme's we would like to mention are reported in [Evelyn Z. McClave, 2000], [Nicole Chovil, 1991], [Loredana Cerrato and Mustapha Skhiri, 2003b] and [Jina Lee and Stacy Marsella, 2006]. This section uses the findings from mainly these references to analyze head movements in a conversation.

In [Heylen, 2006] the term *determinant* is used to describe why certain behaviors occur, and also why they are performed in a certain kind of way. Determinants cover all factors having a role in why a certain action is performed, whether performed using deliberation, to carry out an intention, or whether it's a manifestation of other processes present in communication. In the next section an effort made to find out what behaviors and the interpretation thereof has to do with intentionality.

3.3 Intentionality and interpretation

Since it is not trivial to accept the possibility of reading the mind of a person, as will be practiced later in this chapter, one section of this thesis is devoted to explaining the stance taken here on 'reading the mind' of human subjects. In the introduction of El Kaliouby's report [Rana Ayman el Kaliouby, 2005], she writes:

> Mind-reading or theory of mind is the terminology used in psychology to describe people's ability to attribute mental states to others from their behavior, and to use that knowledge to guide one's own actions and predict those of others [PW78, BRF+96, BRW+99]. It is not, as the word is often used in colloquial English, a mystical form of telepathy or thought reading. The mental states that people can express and attribute to each other include affective states or emotions, cognitive states, intentions, beliefs and desires. An essential component of social intelligence, mind-reading enables us to determine the communicative intent of an interaction, take account of others' interests in conversation, empathize with the emotions of other people and persuade them to change their beliefs and actions.

So, what is this *communicative intend* El Kaliouby writes about? From the view of an agent as an intentional system, as summarized in [M. Wooldridge, 2001, p. 30], it can be concluded that it is sometimes useful to use the *intentional stance* to explain and describe how a system works. Also, from the perspective of analyzing head movements in conversations, and defining models of behavior based on this analysis, it is assumed here that adding information about whether a head movement has a certain intentionally or not could yield some important consequences for the model and the analysis.

Consider the next two descriptions of fragments taken from the GT2M data (see appendix A).

1. At the start of the (forced) small-talk session, Loui and Anke are trying to get the conversation going, so Loui starts by asking "Wie sieht dein Hund aus?". Anke replies with "Mein Hund?". As she does this, her head makes a short and fast movement down and to the side in the direction of Loui. Loui quickly responds with "Ja" and a small nod down. After this Anke starts talking about her dog.

2. In the task-oriented session between Rob and Lidewij, Rob is talking and Lidewij is listening while holding a pen. At one point, while playing with it, the pen makes a tick sound. While still listening Lidewij gazes down to the pen, but very quickly she gazes back at Rob. Rob doesn't notice the tick it seems, and just continues his talk.

From the first fragment we could say that the reply from Anke with the head movement was a question if she had heard and understood what Loui was asking. Another take on this is that Anke heard and understood the question well, but was actually asking something else. To illustrate this, we could translate the interaction into the following:

L: "Do you want to talk about your dog?"

A: "I'm not sure, do I have to?"

L: "Yes, please."

What this 'translation' shows is that the exact words do not necessarily represent the communicative intend; a researcher analyzing this has to look beyond the words to get the meaning and intention of it. In the first interpretation the head movement could be classified as a request for confirmation about understanding. In the second interpretation the head movement is part of a negotiation about turn-taking. Although the two interpretations might be interrelated, the label given to Anke's reply is different. Still, these are just two of many possible interpretations.

The gaze behavior of Lidewij in the second fragment doesn't seem to have any communicative intend involved. The reason Lidewij looks down can be explained by her just checking to see where the sound comes from, or she is asking herself what she is doing. In any case this action was probably not intended to *communicate* something, and it didn't disturb the ongoing conversation.

An interesting point about interpretation and intentionality is made in [Alessandro Duranti, 2006], where intentionality and the use of the concept in language is considered from a cross-cultural point of view, specifically in a comparison of Samoan and English language. Duranti argues that intentionality does not necessarily presuppose consideration, thought, or conscious action by the actor. Instead Duranti, in formulating a more universal sense of the term, characterizes intentionality as an action that has a direction. For example an action can be perceived by the Listener as having a direction towards a certain effect, despite the fact that the Speaker did not 'intend' (in the narrow sense) for the action to have this effect. After summing up some examples of the use of the concept of intention, Duranti writes:

> In all of these cases, we might be able to recognize the 'directionality' of particular communicative acts (e.g. through talk and embodiment) without being able to specify whether speakers did or did not have the narrow intention to communicate what is being attributed to them by their listeners.

Looking back at the first example the problem was that it is difficult to exactly establish which intentions were involved in this interaction. Also this example stresses that context and subjectivity of the observer plays an important role in stating the intention of the actions. But one way or another, intentionality was clearly involved in this type of interaction. Whereas in the second fragment one could argue that Lidewij didn't have the intention of looking at the pen, but was just involuntarily drawn, by instinct, to look at where the sound was coming from. On the other hand, the interpretation could also be that Lidewij was intentionally looking at the pen, because she wanted to ask herself why she was playing with it, or how she made the tick sound.

So, when analyzing behaviors in conversation, taking intentionality into account can make analyzing such fragments as presented above difficult, but at the same time it can tell a lot about what's going on, and what it means. The following distinction sketches the different levels of intentions of head movements that are distinguished here.

3.3.1 Deliberate

The person uses, usually goal-directed, cognitive processes to activate the behavior. The behavior is performed with the intention (narrow view) to manipulate the environment. For example if someone performs a head shake, one may interpret that as an intention to manipulate the belief of the other person about it's (negative) stance towards the point being discussed at that moment.

3.3.2 Automatic

The person performs behaviors, possibly subconsciously, and seems to do this without using processes of the mind. Intentionality in the sense formulated by Duranti [Alessandro Duranti, 2006] of the behavior *is* present, though not before the action took place. Only after the action the performer and the observers perceive it as directed towards a certain effect. A simple example of this type of behavior is the second fragment shown above, where Lidewij is just looking to the pen because she heard some sound.

A more complex example, which may be related to the term *dead-reckoning* which Thórisson uses in [Kristinn R. Thórisson, 2002] for a type of planning and preparing a response. To illustrate: if a listener predicts that the speaker will soon end his sentence, a back-channel signal such as a nod may already be prepared to be performed if everything goes as expected.

3.3.3 From reflexes or other physical needs

Sometimes a person moves unwillingly, directly following a stimulus received from the environment. In this case intentionality is not present in the behavior in the sense of the formulation used by Duranti; the behavior has no direction, it is involuntary, it is not part of social interaction. Classical examples are coughing, sneezing and blinking to wet the eyes.

3.3.4 Intentionality in analyses

In the next section an analysis is made from the use of head movements in conversations from the SAL videos. In this analysis behaviors are studied from two angles (1) what do they look like, (2) what lies at the basis of their intentions. When considering what has been touched upon in this section, this second angle stresses the assumption that the behaviors that will be analyzed have intentionality. In other words, when interpreting the head movements, it is assumed that these behaviors have a directionality, in Duranti's terms, and that by labeling these behaviors, their 'direction' is identified. The distinction as presented here is not explicitly made in the analysis shown next, nonetheless it is believed here that having a clear notion of intentionality will support the motive of trying to find the why behind the use of head movements.

3.4 Classifying head movements

This section presents the use of a framework which enables the classification of head movement behavior of a Speaker in a dyadic face-to-face conversation. The analysis performed here has it's foundations in the ideas expressed in this chapter so far. The *framework* incorporates these ideas in the sense that:

1. a clear distinction is made between form and function/determinant,

2. it integrates different views on the use of head movements, and

3. it uses a flexible (data) structure suitable for use in computational models, enabling an ECA to automatically interpret and classify Speaker behavior.

Here a concrete example of the use of such a framework is presented. First in subsection 3.4.1 the online head movement database tool is presented, which allows for the analysis done here to be accessible for review. Then, in section 3.4.2 the head movement segments and the behavioral descriptors are shown. These fragments, taken from the SAL database, are the basis of the data set. In subsections 3.4.3 to 3.4.6 some findings from other research on specific categories of head movement determinants and the label set used in the analysis are described. Subsections 3.4.7 and 3.4.8 show the results of the analysis on the data set.

3.4.1 Head movement database

Data from the SAL database is analyzed and functions of head movements are linked to their form. To give insight into the data from the analysis an accessible, online head movement database was constructed. This database contains manually selected head movement segments, and combines automatic descriptions of head movements using the head tracker presented in section 2.2, as well as manual annotations of head movement form and function. The data was gathered using the following strategy:

1. Manually select segments of the video data in which a head movement is observed and label it using the label set presented in subsection 3.4.2.

2. Annotate each of the head movements with one or more head movement determinant (defined in subsections 3.4.3 to 3.4.6).

3. Select a subset of determinants that are most common and keep only head movement segments having these determinants.

4. Use the head tracker on the data and combine this with the manually selected segments with their annotations.

This recipe is applied to the first three minutes of three persons chatting with the SAL character Prudence. The result is a browsable, semi-interactive database which can be accessed on-line[1]. Figure 3.1 shows a snapshot of this tool.

The upper-left part of the canvas shows a summary of the annotations that apply to the selected fragment. The annotations have a category label, i.e. a functional label that applies to this fragment. A 'context' description of the utterance, or other non-head movement behavior. It also shows other categories that apply to the selected segment, a head movement description using the behavioral descriptors defined in subsection 3.4.2, an automatically generated description derived from the MOVETOMOVE program (see subsection 2.4.3), and optionally some comments.

The upper-right part shows the list of segments by category. The lower-left part contains the segment from original video data. The lower-right part is RUTH mimicking the head movement using the output from the tracker as input. There two modes for this: *hlm* shows the movements based on the tracker output as defined in section 2.4.3, *hrot* shows RUTH directly using the rotation angles output from the HEADROT program (see subsection 2.2.3).

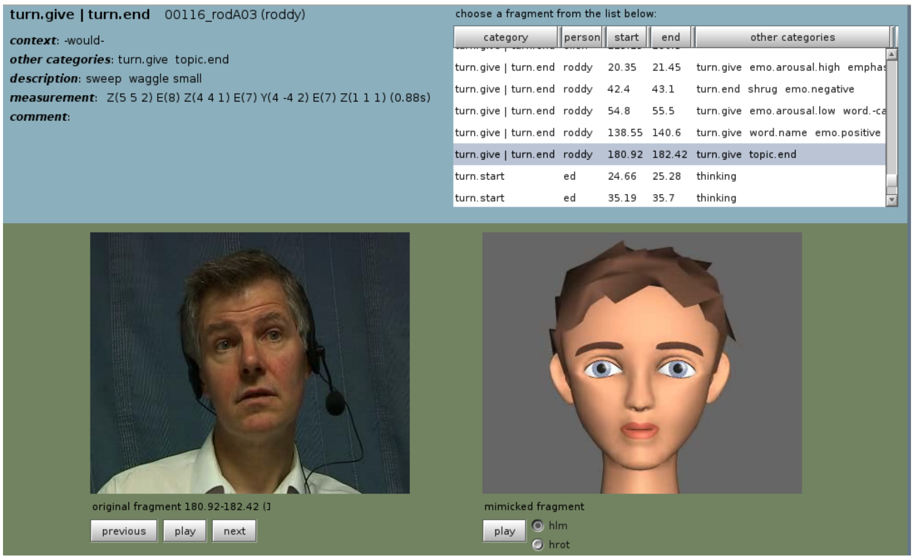
3.4.2 Behavioral descriptors

The manual annotation is performed on three aspects of a head movement for which table is 3.1 used. As concluded in section 2.7, to accurately describe

Table 3.1: elementary movement types and modifiers

movement	*description*	*direction mod*	*shape mod*
nod	relatively fast movement up and/or down	fast	up
shake	relatively fast movement from side to side	slow	down
move	steady (slow) movement in one direction	side	jerk
sweep	steady fast movement to the side	repeated	small
roll	head roll		big
waggle	small movements back and forth		arch

[1] see SAL head movement database http://wwwhome.cs.utwente.nl/~rkooijma/hmdb.swf



turn.give | turn.end 00116_rodA03 (roddy)

context: -would-
other categories: turn.give topic.end
description: sweep waggle small
measurement: Z(5 5 2) E(8) Z(4 4 1) E(7) Y(4 -4 2) E(7) Z(1 1 1) (0.88s)
comment:

choose a fragment from the list below:

category	person	start	end	other categories
turn.give | turn.end	roddy	20.35	21.45	turn.give emo.arousal.high emphas
turn.give | turn.end	roddy	42.4	43.1	turn.end shrug emo.negative
turn.give | turn.end	roddy	54.8	55.5	turn.give emo.arousal.low word.-ca
turn.give | turn.end	roddy	138.55	140.6	turn.give word.name emo.positive
turn.give | turn.end	roddy	180.92	182.42	turn.give topic.end
turn.start	ed	24.66	25.28	thinking
turn.start	ed	35.19	35.7	thinking

original fragment 180.92-182.42 (J

mimicked fragment

● hlm
○ hrot

Figure 3.1: snapshot of corpus browser

a head movement it is useful to take aspects like speed, magnitude, direction into account. Here, a transcription of head movements also capturing these subtleties includes three dimensions as listed in the table. The first dimension is the movement description, specifying the main characteristic of the movement. These movement types are based on findings presented in [Heylen, 2006] and [Loredana Cerrato and Mustapha Skhiri, 2003a]. The other dimensions are referred to as *modifiers*. The direction modifier is about speed, direction and cyclicity of the movement. Some of the main movement types have already some of these characteristics contained in their description: in these cases the modifier does not apply. The shape modifier is about the direction, magnitude and progress of the movement over time. For example 'move side arch' describes a lateral movement to one side with an arch-like trajectory. To illustrate: if we would track the point of the nose, as did Heylen in [Dirk Heylen, 2005a], then figure 3.2 would show the trajectory of the tracked point as an arrow. However,

Figure 3.2: trajectory of *move side arch*, as an arrow tracing the nose point

this kind of illustration of the movement cannot show e.g. if the movement was fast or not, hence the construction of the head movement database (3.4.1) were each movement can be shown in motion.

Speaker head movement segments in the data

Following step 1 (see above) approximately 500 segments were manually selected. Some basic statistics about the data and the segments can be found in table 3.2. The terms *multiple* and *single* HM as used in the table, refer to whether the segment has multiple head movements, or not. Note that Roddy

Table 3.2: basic fragment statistics

	all	*Roddy*	*Ed*	*Ellen*
duration of fragment	7m38s	3m21s	2m07s	2m10s
HM segments	520	165	178	177
single HM count	97	42	33	22
avg. duration single HM (s)	0.83	0.83	0.81	0.86
multiple HM count	29	9	9	11
avg. duration multiple HM (s)	2.59	2.88	2.85	2.13

has about the same amount of segments in a longer time, this anomaly may be an idiosyncrasy of Roddy or a shift in how the manual segmentation was performed. Note that due to the exploratory nature of this exercise only one annotator was involved in the segmentation and labeling process, reliability of the analysis can therefore not be established. This also goes for the other annotations on the data.

3.4.3 Cognitive processes

The way our thinking works is an important determinant for explaining deliberate behavior. Cognitive processes involved reported e.g. by Newell [Allen Newell, 1990] are: memory, recognition/understanding, learning/adaption, repairing/correcting, deciding/choosing and planning. In this work Newell takes a modularist stance; the human mind is assumed to be constructed from 'devices' or 'modules' that make up the brain machine. What are these modules, how do they work, and how do they affect our analysis and design choices for making a SAL?

One of the things important for the design of SAL is that our brain puts certain *restrictions on the communication process*. These restrictions cover aspects like timing and synchronization, which are important characteristics in human communication; not all types of behavior are processed with the same priority or speed. Thórisson proposes in [Kristinn R. Thórisson, 1999] that, when designing agents for real-time interaction, one has to capture the timing relationships between the different components of cognition.

In the Ymir architecture presented in his work, three layers are distinguished: content, process control and reactive. In the content layer domain-specific knowledge is available, this layer is responsible for the *what* in the conversation. The process control layer contains knowledge about social interaction and handles conversational processes such as turn-taking. The reactive layer is about perception and performance of simple behaviors and makes this information available to higher layers. Timing aspects are the focus of the Ymir architecture. For example the content layer, which hosts rules for knowledge and understanding has a lower priority and updating frequency than the reactive layer.

The views presented above stress the internal workings of the human mind and takes this as a starting point for making design decisions. Another important issue however, is the observation that humans seem to express their *cognitive status*. For example the 'thinking face' [Dirk Heylen, 2005b, Nicole Chovil, 1991, M.H. Goodwin and C. Goodwin, 1986] is not only a symptom of mental processes, but can also be an intentional sign to express that one is thinking, and thus function as a request to get some more time to react.

So, uncovering cognitive processes and making them explicit is not only important because of design decisions, but also addresses the observation that people will (deliberately or not) express them and could thus bear meaning. Other behaviors, expressed through specific gaze patterns are word search [Dirk Heylen, 2005a, Evelyn Z. McClave, 2000] and lexical repairs [Evelyn Z. McClave, 2000] and speech corrections [Nicole Chovil, 1991].

labels in the cognitive category

The following lists the labels that will be used in the analysis in the COGNITIVE category. Note that they all relate to the thinking process.

- REPAIR: The Speaker made a mistake in his utterance or is stuttering. The Speaker continues speaking, but is correcting pronunciations, sentence formulations, word choice et cetera, by replacing them. Might be accompanied with a short "no", or "I mean...". Expected head movements reported in literature: shake
- WORD-SEARCH: The Speaker is doing a lexical retrieval, i.e. searching for the words to utter next. The difference with REPAIR is that the Speaker is silent for a moment, or fills this pause with "uh" or something similar. This is reported to be accompanied with specific gaze patterns, such as the eyes moving quickly from side to side. Expected head movements: move side fast, shake.
- THINKING: The Speaker is thinking about something, often after a request from the conversational partner, or after a topic end. This label applies to a segment when it does not make sense to label the segment as REPAIR or WORD-SEARCH. Expected head movements: roll, move side, move up.

3.4.4 Social determinants

When viewing conversations as social activities, the *communicative acts* [Stefan Kopp et al., 2006] involved in this activity are somehow coordinated. In [Dirk Heylen, 2005b] Heylen explores the functions and determinants of head movements mainly in the context of *language as social action*. Adopting this view, one could argue that many functions and determinants of head movements as used in conversation can receive the label 'social'. Based on this and other work a short-list is presented here of social functions and determinants. Whether these all relate to head movements remains to be investigated, but it cannot beforehand be ruled out that they all have some influence on the way head movements are performed.

A distinction can be made between aspects that (1) influence how head movements are performed and perceived, and (2) play a role in the communication process that have more to do with how people cooperate and how conversations are managed, coordinated. Some examples of *(1) variables of influence* are: affect, familiarity, dominance, culture, face, social protocols/conventions, dependence, status/power, adaption, attitude, engagement, mood, approach, rapport, compassion, character, personality. (2) Social *processes*: mirroring, back-channeling, feedback, synchronization, turn-taking, theory of mind.

labels in the social category

In the analysis presented here we focus on *attitude*. Attitude can mean a lot of things, and can be argued to be related to a wide variety of other determinants. The common factor of attitude as meant here, is that it denotes a stance towards some other person, acts, things et cetera. The following labels will used in the annotations:

- LIKE, DISLIKE: The Speaker expresses a positive or negative stance towards some thing, person, or what is being said at the moment. Behaviors for these two labels are expected to be related to respectively positive or negative associations, so typical head movements expected are: nod and shake.

- CERTAIN, UNCERTAIN: The Speaker expresses that he/she is not certain of what is being said at the moment. Uncertainty is reported by Evelyn Z. McClave [2000], to be accompanied by a lateral shake.

3.4.5 Linguistic and communicative

Analyses of verbal expressions can be performed at different granularities of the *speech unit* (or *lexical item*). These granularities include e.g. syllable, word, clause, utterance, sentence, dialogue et cetera. In [Nicole Chovil, 1991] Chovil notes about the transcript used in her analysis:

> Some [facial] displays occurred amid only one word, whereas others were held for a clause or the entire utterance. Other displays occurred before the utterance began, after the utterance was finished, or in the absence of any spoken content.

So co-occurrence of facial display and lexical items, is not always timed exactly, or not correlated at all. There are however cases where facial display, and head movements as well, *do* co-occur with lexical items of different granularities. For example the performance of a head nod can be timed exactly with the word “yes”, but also a head shake can be sustained during the whole sentence “I don’t like this situation one bit.”.

In the analysis of head movements in relation to the *content* of the conversation: the *what* in the conversation, we will refer to this level of analysis here as the *content level.* Examples taken from [Justine Cassell, 2000] that can be identified at this level are *theme/rheme*, or whether the utterance contains *given* or *new* information. Another example is whether the speaker is using *portrayal* to denote that the content presented should be interpreted specially within the context of an event that happened earlier, or when mimicking another person [Nicole Chovil, 1991].

labels in the content level

For the content level analysis presented here, labels are mostly taken from typologies used in [Nicole Chovil, 1991], [Evelyn Z. McClave, 2000] and [Jina Lee and Stacy Marsella, 2006]. The following labels are defined in the content level:

- AFFIRMATION: Includes also statements of agreement and understanding. Usually expressed by the Speaker at the start of his/her turn. Expected head movement: nod.
- INTENSIFICATION: Reported movements by McClave are shakes co-occurring with words like: “exactly”, “very”, “a lot”, “great”.
- EMPHASIS: Head movements of the Speaker that co-occur with a prosodically marked (stressed) word. Reported to be accompanied by brow raising. Expected head movements are: nod down short.
- LIST: The Speaker lists several possibly opposing items. Head movements reported are small movements to the side that co-occur with each item, possibly moving from one side to the other at each item.
- INCLUSIVITY: In the work of McClave this label is used mostly on lateral sweeps, co-occurring with words like “everyone”, “everything”, “whole”.
- DEICTIC: Often a gesture of the head towards a specific point in space, also called *referential use of space*. The term *space* must be interpreted in this context as widely as possible; i.e. imaginary/abstract space is also included. The movement of the head is dependent on the position of the object in ‘space’. Expected is at least the head movement roll short fast, possibly with a small up component. An example of this commonly practiced movement co-occurs often with words like “that” or “back then”.
- WORD: Applies to head movements of the Speaker co-occurring with a word, and expressing the meaning of this word itself. This label is also used for head movements expressing the meaning of a word that is not uttered but only expressed through the movement. One example is a shake co-occurring with the word “no”. Chovil also found facial displays of “yes”, “not” and “but”.
- SHRUG: Possibly involves shoulder and brow raising, expected movement is roll.

discourse level

Another level of analysis related to linguistic and communicative determinants distinguished here is the *discourse level*. At this level people communicate about the structure of the conversation. It is about the semantic relations between sentences and parts of sentences. At this level determinants like speech act, performative, dialogue act et cetera, play a role.

The following list contains the labels distinguished in the analysis. The terminology for the *clarification* and *explanation* labels come from [Nicole Chovil, 1991], although in her data these labels were not often used. The *quote* labels are from [Evelyn Z. McClave, 2000]. The *topic* label and sublabels were inspired by multiple sources [Catherine Pelachaud, 2002, Evelyn Z. McClave, 2000, Pelachaud et al., 1998, Yuri Iwano et al., 1996, Nicole Chovil, 1991].

- CLARIFICATION and EXPLANATION: The Speaker is elaborating on something he/she was referring to earlier.

- QUOTE: *direct quote* (McClave) marks a shift from indirect to direct discourse. The Speaker uses somebody else's words or may even do a portrayal (Chovil) of somebody. Expected movement is: move side.

- TOPIC: The Speaker is talking about the *content* (as used above) of the conversation, but this content can be structured; e.g. a new topic can be introduced by the Speaker, the Speaker can side-track and then come back to the topic et cetera. The term *narrative* is closely related to *topic* because both focus on the *content* of the talk, and how the content is structured over time. The following sublabels are used in the analysis:

  – START: A new topic is started. *Story announcement* in the terms of Chovil, co-occurs with brow-raising and looking up. Coincides possibly but not necessarily with the start of a turn.

  – MARK: Has parallels with *story continuation* in the terms of Chovil. The Speaker makes an important point in the topic being discussed, possibly after a side-track. This is marked according to Chovil by words such as “so”, “but”, “then”, “so anyways” and brow raising.

  – SHIFT: A topic shift is like a topic mark only the Speaker now makes a major shift in the topic being discussed so-far. The Speaker starts opposing, or complementing what was said earlier, or a whole new main topic is started implicitly. The difference with the use of topic shift compared to subsequent end and start, is that with a topic shift the transition to another topic is not so sharp, but more fluent.

  – COMMENT: The Speaker makes a side-track in his story. The main topic is still prevalent but another (sub)topic is shortly commented upon.

  – END: *end of story* (Chovil). The Speaker makes it clear that he/she finished talking about the main topic. May coincide with the end of a turn.

interaction level

The *interaction level* contains labels that deal with the interactive nature of conversation, and how people coordinate their behaviors during conversation. Examples of interactional processes in conversations are turn-taking and backchanneling, as studied in e.g. [Evelyn Z. McClave, 2000, Nicole Chovil, 1991, Starkey Duncan jr, 1975], and grounding [Yukiko I. Nakano et al., 2003]. In chapter 4 the interactive processes are studied in more detail, and the terms *turn* and *floor* will be explained. Only turn-taking is taken into account here. The following label and sublabels are used:

- TURN-TAKING

  – END: The Speaker is finished talking and keeps silent for a moment.

  – GIVE: The difference from END is that the Speaker explicitly hands over the floor to the conversational partner, often after doing a request or asking a question.

  – START: The Speaker just got the floor at starts talking.

  - KEEP: The Speaker has the floor, and makes it explicit that he/she wants to keep it, although the Listener interrupts, or wants to interrupt.

syntactic determinants

The occurrence of head movements also seems related to grammatical or syntactical features of speech. For example in [Nicola Cathcart et al., 2003] a model incorporating trigram part-of-speech frequencies is used to predict backchanneling behavior. Determinants from this category are however not taken into account the analysis presented here.

3.4.6 Emotion

Emotional head movements co-occur with certain facial or bodily behaviors like smiling or frowning, also with certain spoken content or speech features like high or low pitch, faster or slower speech (see e.g. [Hans Peter Graf et al., 2002]), and specific gaze patterns (see e.g. [Dirk Heylen, 2005a] for some examples)

An approach to analyze behaviors related to emotion, is mentioned in [Roddy Cowie et al., 2005]. In this approach the Feeltrace tool is used by annotators to continuously classify the emotions of participants talking to a SAL character. Emotions are classified on two emotion dimensions: activation (arousal), evaluation (valence) [2]. In the analysis presented here arousal and valence are also taken into account, although not in the continuous 2D scale of the Feeltrace tool.

labels of emotion

The head movements of the Speaker are labeled with an emotion label only if the movement seems related to emotion. This means that if arousal or valence is assessed as being neutral at the time, no emotion label is given to the head movements:

- AROUSAL
  - HIGH: When the Speaker is upset he/she may also have high pitched speech.
  - LOW: The Speaker talks slowly, speech pitch is neutral or low.
- VALENCE
  - POSITIVE: The Speaker may be smiling, laughing.
  - NEGATIVE: The Speaker may be frowning, mouth corners may be pulled down.

[2] see Feeltrace tool URL http://www.dfki.de/~schroed/feeltrace/

3.4.7 Label counts in the data

The functional categories and their labels and sublabels presented above were applied to the head movement fragments selected in subsection 3.4.2. From the 500 fragments, and following the recipe of constructing the head movement database (see subsection 3.4.1), the fragments were annotated. It became clear that not every movement could be easily classified in one or category, also some labels were never or almost never used. Also the difference between turn give or turn end was not clear so those were put together. The same goes for explanation and clarification.

After filtering the data has a total of 118 fragments. The total count, the count per person and the average duration per category label and sublabel are listed in table 3.2. The table shows that the labels thinking, emphasis, intensification, topic mark and turn give or end were common: they apply to 10% or more of the fragments. The labels repairing, affirmation and inclusivity were not so common: 5 or less fragments. The average duration varies from half a second to three. Most noticeable is that the fragments labeled in the emotion category have a considerably longer average duration, about three seconds, than that of other categories.

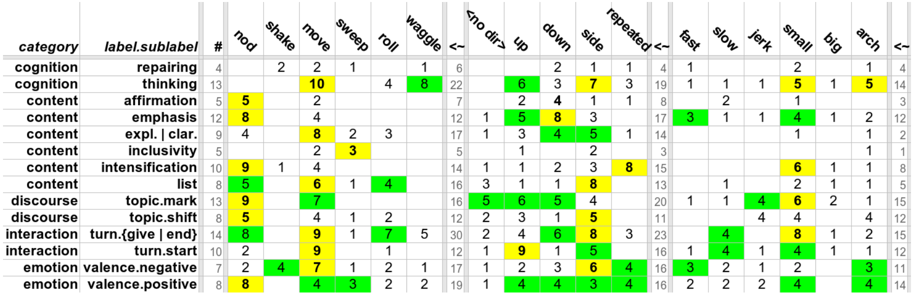
Table 3.3: basic category statistics

category	*label*	*sublabel*	*frags*	*avg dur.*	*Roddy*	*Ed*	*Ellen*
cognition	repairing	-	4	1.01	2	0	2
cognition	thinking	-	13	1.68	4	5	5
content	affirmation	-	5	1.71	3	2	0
content	emphasis	-	12	0.65	5	3	4
content	expl. or clar.	-	9	0.73	3	4	2
content	inclusivity	-	5	0.58	4	1	0
content	intensification	-	10	0.97	4	4	2
content	list	-	8	0.9	3	3	2
discourse	topic	mark	13	0.84	5	4	4
discource	topic	shift	8	0.86	4	2	2
interaction	turn	give or end	14	1.19	5	5	4
interaction	turn	start	10	0.64	4	4	2
emotion	valence	negative	7	3.09	3	1	3
emotion	valence	positive	8	3.03	2	4	2

3.4.8 Mapping functions to behaviors

As promised, now a look is taken at the link between functions of behaviors and their form. To get an idea of what kind of analyses the presented data allows, we show figure 3.4. The table shows observed head movement aspects mapped to categories assigned to segments containing head movements. The numbers represent the co-occurrence counts of function and elementary movement. The yellow marked cells indicate the highest counts, the green the second, or third highest counts. If we compare the VALENCE.NEGATIVE and the VALENCE.POSITIVE categories with each other it can be seen that the first co-occurs head movements typed move with modifier side, while the latter co-occurs mainly with

Table 3.4: mapping of functions to elementary movements



category	label.sublabel	#	nod	shake	move	sweep	roll	waggle	<~	<no dir>	up	down	side	repeated	<~	fast	slow	jerk	small	big	arch	<~
cognition	repairing	4		2	2	1		1	6			2	1	1	4	1			2		1	4
cognition	thinking	13			10		4	8	22		6	3	7	3	19	1	1	1	5	1	5	14
content	affirmation	5	5		2				7		2	4	1	1	8		2		1			3
content	emphasis	12	8		4				12	1	5	8	3		17	3	1	1	4	1	2	12
content	expl. | clar.	9	4		8	2	3		17	1	3	4	5	1	14				1		1	2
content	inclusivity	5			2	3			5		1		2		3						1	1
content	intensification	10	9	1	4				14	1	1	2	3	8	15				6	1	1	8
content	list	8	5		6	1	4		16	3	1	1	8		13		1		2	1	1	5
discourse	topic.mark	13	9		7				16	5	6	5	4		20	1	1	4	6	2	1	15
discourse	topic.shift	8	5		4	1	2		12	2	3	1	5		11			4	4		4	12
interaction	turn.{give | end}	14	8		9	1	7	5	30	2	4	6	8	3	23		4		8	1	2	15
interaction	turn.start	10	2		9		1		12	1	9	1	5		16	1	4	1	4	1	1	12
emotion	valence.negative	7	2	4	7	1	2	1	17	1	2	3	6	4	16	3	2	1	2		3	11
emotion	valence.positive	8	8		4	3	2	2	19	1	4	4	3	4	16	2	2	2	4		4	14

nods. More interesting results from the table are e.g. the co-occurrence of TURN.START with MOVE.UP and THINKING with MOVE.SIDE. These kinds of results, when more thoroughly investigated, could help in automatic recognition of the meaning of head movements in a certain context. Using the knowledge from this kind of analysis, we could pose that for example if an artificial listener observes a move to the side there is a high probability that, given other contextual information, the speaker is thinking. It could also help with improving recognition of behaviors in other modalities. For example if a small and fast nod down is given (EMPHASIS) speech recognition and understanding modules could use this information to their benefit.

3.5 Conclusion

The following list summarizes some of the links between form and function found in the analysis presented in this chapter:

- The combination of NOD DOWN FAST, could be correlated to the Speaker emphasizing what is being said.
- Head movements in a negative emotional context are mostly MOVE SIDE, while in positive emotional contexts NOD occurs more.
- While the Speaker is THINKING, MOVE SIDE is common.
- At turn starts the Speaker uses MOVE UP.

Drawing on theories about cognition, social behavior, emotion, linguistics during face-to-face communication, it is shown that a single framework of interpretation and classification of speaker behavior can be constructed. It is then possible to derive correlations between form of head movement behavior with functions of this behavior. The broad view on the different theories shortly touched upon in this chapter, also shows the broad spectrum in which head movements could be interpreted. This paves the road for the first steps towards automatic interpretation of Speaker head movements.

The next step for designing a SAL is then to use rules from the knowledge gained in analyses such as presented here, in a decision module responsible for the generation of Listener behavior.

Chapter 4

Interactive behavior in conversations

One of the goals pursued in this thesis is to explore some models of conversational behavior of a Listener. In this chapter we take an in-depth look at the various aspects that are involved in *interactive behavior* employed in a conversation. The term *interactive behavior* is used here to denote the behaviors that conversants use to communicate and coordinate the communication process. This section will address the following questions:

- which interaction patterns of conversational behavior occur in face-to-face conversations?

- do conversants use specific behaviors to interact with each other in specific contexts?

The first subsection of this chapter 4.1, gives some examples of model of interactive behavior found in literature. The examples function to highlight the processes that will be explored in this chapter. The transcribed behaviors are: utterances, speech prosodic accents, gaze and blinking and head movements. The second subsection 4.2 shows the data collection and the transcriptions of behaviors in this data. Also, some basic statistics and graphical representations of the transcription data is shown here. The next subsection 4.3 shows annotations made on the transcriptions and some basic statistics and graphical representations thereof. The annotations deal with interactive processes of turn-taking, feedback and mirroring. In section 4.4 a method for analyzing the transcriptions of the behaviors and the annotations of the interactive processes is presented. The method aims to enable detailed analyses of interactive processes by counting co-occurring behaviors. How to use the method on the annotations and transcriptions and the parameters involved in the analysis are shortly reviewed in section 4.5. In this section also some examples are given of rules that can be deduced form the results of the analysis. Section 4.6 performs more specific analyses of behaviors in interactive contexts.

4.1 Rules of interactive conversational behavior

It is assumed here that a structured way of investigating interactive behavior in human-human conversations, will enable the production of a rule-base that can be used to *artificially* produce *natural* listener behavior. The examples shown in this section, taken from literature, use rule-based models of interactive behavior, they (1) show how behaviors function in interactive contexts, thus mapping behaviors to conversational functions and (2) give insight into computational models capable of generating behaviors for use in embodied conversational agents. The examples are presented here to show three different processes important for the design of SAL that deal with interaction: feedback, turn-taking and mirroring.

4.1.1 attention and interest

In [Christopher Peters et al., 2005a] focus is on researching how behaviors of an artificial agent influences the level engagement ascribed to the agent by the (human) conversational partner. The authors mention the following processes as capabilities for an Addressee to be perceived by the Sender as an engaged ECA: attention, perception, comprehension, internal reaction, decision and generation. Next it is argued that attention and interest are vital for the impression of engagement; the Addressee uses signals of attention and interest to let the Sender know that it is still has a certain level of engagement in the conversation. The Sender picks up these signals and may interpret them as intended. To show how such an interactive process can be implemented three algorithms are presented. The first shows how one agent can estimate the intention of the other to start a conversation based on attention and interest level. The level of attention and interest is inferred from a visual perception system that gives information about eye, head and body directions. The second and third update the probabilities of state transitions in finite state machines controlling the behaviors of the Sender and Addressee based on gaze behavior of both agents. The three algorithms show that knowledge about how behaviors function, e.g. how gaze durations and their distribution influences perception of attention and interest, can be used to construct models governing the behavior of ECAs.

4.1.2 turn-taking

Thórisson shows in [Kristinn R. Thórisson, 2002] a turn-taking model, the Ymir Turn Taking Model, consisting of three layers: content, process control and reactive layer (see also 3.4.3). To meet the requirements of real-time decision making the layers are processed in parallel and have different responsibilities and priorities. To give an example of the rules involved in the turn-taking process, here a state transition rule is presented from the turn-taking model.

```
transition(other-has-turn, i-take-turn) IFF:
(AND
(Time-since(Other-is-presenting) > 70 msec)
(Other-is-giving-turn = T)
(Other-is-taking-turn = F)
(OR
(Others-intonation-going-up = T)
(Others-intonation-going-down =T)))
```

What's interesting about this rule is that it combines more complex perceptual inputs (OTHER-IS-PRESENTING, OTHER-IS-GIVING TURN and OTHER-IS-TAKING) with more low-level percepts (OTHERS-INTONATION-GOING-UP and OTHERS-INTONATION-GOING-DOWN). What we would like to find out is (1) if percepts/signals like OTHER-IS-GIVING-TURN can be deduced from specific behaviors and (2) if these signals are actually present during turn-transitions in real conversations.

4.1.3 mirroring

Maatman focuses in [R.M. Maatman, 2004] on mirroring and modelling responsive behavior for listening agents. The research does not present a unified view on conversational behavior, cognitive models, social interaction et cetera, but is does show and experiment with some rules that might be useful for modelling listeners especially their mirroring behaviors. The next rules present a selection of the rules that are about mirroring:

```
if the human performs a posture shift then mirror this posture shift
if the human performs a head shake then mirror this head shake
if the human performs major gazing behavior then mimic this behavior
```

Although the research does not evaluate these rules thoroughly, it would be interesting to see if we can find evidence of these rules in the video data studied here.

In the remainder of this chapter we will focus on three *interactive processes* of conversational behavior:

- turn taking
- feedback
- mirroring

Data from the GT2M data (see section 4.2) as well as insights from other research are used to give an analysis of these processes. Most attention goes out to *behaviors and aspects of behaviors* of:

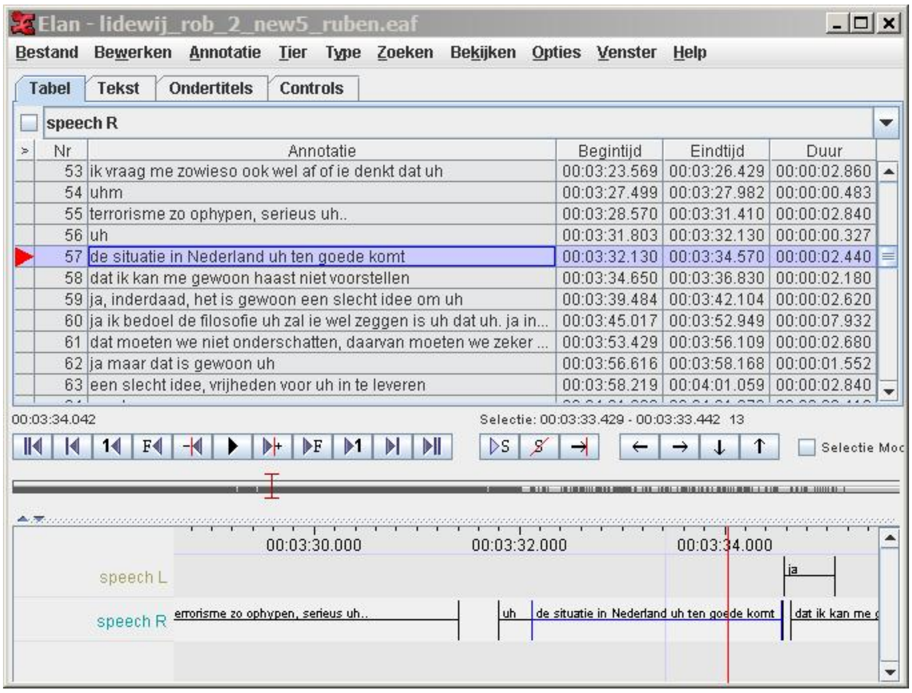
- gaze and blinking
- head movements
- speech

First the data collection is presented, which gives an overview of what the data constitutes, how it was collected, and possible ways to look at different aspects of the raw data.

4.2 data collection

Recorded videos from dyadic face-to-face conversations were selected from the GT2M data in which the participants were asked to work together on the task of formulating three questions for prime minister Balkenende. The participants have to cooperate to come up with the questions to be written down on paper. More details of the GT2M data can be found in appendix A.

The first step we take for analyzing interactive behavior in these conversations is to *observe* and *register* what is happening in the video. Since the video fragments have both a visual and an audio component, transcriptions can be made of bodily movements as well as speech. The observations are captured using the ELAN annotation tool [Language Archiving Technology, 2007]. Figure 4.1 shows a screen-shot of an utterance annotation in ELAN.


Elan - lidewij_rob_2_new5_ruben.eaf
Bestand Bewerken Annotatie Tier Type Zoeken Bekijken Opties Venster Help
Tabel Tekst Ondertitels Controls
speech R

>	Nr	Annotatie	Begintijd	Eindtijd	Duur
	53	ik vraag me zowieso ook wel af of ie denkt dat uh	00:03:23.569	00:03:26.429	00:00:02.860
	54	uhm	00:03:27.499	00:03:27.982	00:00:00.483
	55	terrorisme zo ophypen, serieus uh..	00:03:28.570	00:03:31.410	00:00:02.840
	56	uh	00:03:31.803	00:03:32.130	00:00:00.327
▶	57	de situatie in Nederland uh ten goede komt	00:03:32.130	00:03:34.570	00:00:02.440
	58	dat ik kan me gewoon haast niet voorstellen	00:03:34.650	00:03:36.830	00:00:02.180
	59	ja, inderdaad, het is gewoon een slecht idee om uh	00:03:39.484	00:03:42.104	00:00:02.620
	60	ja ik bedoel de filosofie uh zal ie wel zeggen is uh dat uh. ja in...	00:03:45.017	00:03:52.949	00:00:07.932
	61	dat moeten we niet onderschatten, daarvan moeten we zeker ...	00:03:53.429	00:03:56.109	00:00:02.680
	62	ja maar dat is gewoon uh	00:03:56.616	00:03:58.168	00:00:01.552
	63	een slecht idee, vrijheden voor uh in te leveren	00:03:58.219	00:04:01.059	00:00:02.840

00:03:34.042

Selectie: 00:03:33.429 - 00:03:33.442 13

Figure 4.1: utterance tiers in ELAN

As mentioned, our main focus is on behaviors of eye-gaze, head movements and speech. The transcriptions register these behaviors with a reasonable amount of detail. More detail would be too time-consuming and also it might produce too much information for the analysis process. Less detail restricts the analysis process in that there may be not enough information. So the annotation schemes presented here try to find a middle-ground between too much and too little information.

4.2.1 utterances

The utterance annotations were made by defining two *tiers* in ELAN, each for one participant, and carefully segmenting them into fragments containing an utterance. The segmentation is done such that:

- no utterance fragment contains a speech pause.
- a sentence start is a new utterance fragment start
- continuing speech after a disfluency or on a repair is a new utterance fragment start.

In this way a sentence may span several utterance fragments, or an utterance fragment may contain multiple sentences if the sentences are uttered without a pause or disfluency.

4.2.2 eye gaze direction and blinking

Eye gaze direction and blinking behavior of each participant is annotated in a similar fashion. The big difference is that for this tier a, so-called, *controlled vocabulary* is defined in ELAN. The labels in the controlled vocabulary are shown in table 4.1. Note that *gaze* and *blink* segments are contiguous since a person always has an eye gaze direction or has his eyes closed due to blinking.

At	*gaze at partner*
Blink	blink
R	look to the right
L	look to the left
U	look up
UL	look up-left
UR	look up-right
D	look down
DL	look down-left
DR	look down-right
C	look straight ahead

Table 4.1: eye gaze labels (from the participant's perspective)

4.2.3 head movements

Based on a scheme used in [Hans Peter Graf et al., 2002] a controlled vocabulary is defined for head movements. As in the work of Graf, the annotation scheme defined here focuses on (rigid) head *movements*, rather than head *orientations*. A movement has a starting point and end point in time; fragments in the movie containing (almost) no head movement are not labeled. So it is assumed, as in the work of Graf, that a head movement segment is the part between the start and the end point. The head movement tiers used in ELAN are therefore non-contiguous (a property of the segmentation process in ELAN), as opposed to e.g. the gaze tier.

The label set defines three forms of movement and three axes around which the movements can take place. The '/' labels (abrupt swing in [Hans Peter Graf et al., 2002]) are used here also for movements around an axis that develop in one direction, not necessarily abrupt. The orientation of the axes is shown in figure 4.2. Table 4.2 lists the labels used in the annotations.

Table 4.2: head movement labels

ˆx	nod (around x-axis)
~x	nod with overshoot
/x	movement around x-axis
ˆy	shake (around y-axis)
~y	shake with overshoot
/y	movement around y-axis
ˆz	left-right, or right-left roll
~z	ˆz with overshoot
/z	movement around z-axis

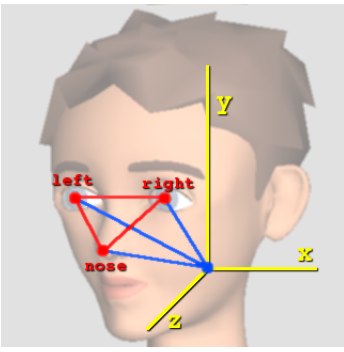


Figure 4.2: orientation of rotation axes

4.2.4 speech prosodic accents

Besides the literal utterance, another feature of speech is annotated. A *speech prosodic accent* refers here to the presence of prominent speech prosody in the audio channel. The main focus here is not to capture lots of subtleties or a wide range of possible prosodic features, like e.g. the ToBi system does[Silverman et al., 1992], but more basic, like stressed words or syllables that really stick out because of drawl, high amplitude or strong pitch change.

4.2.5 three and a half minutes of Rob and Lidewij

In total the GT2M data consists of four small-talk sessions and four task-oriented conversations of approximately ten minutes per conversation. However, here only the first three and a half minutes of one task-oriented conversation is transcribed and analyzed with the schemes presented above. Now some global characteristics of the transcriptions are given.

utterance fragments

In the fragment 104 utterance fragments were identified, 57 from Rob and 47 from Lidewij. The duration of one utterance is on average 1.7 seconds, 1.8 seconds for Rob and 1.5 for Lidewij. The total speaking duration is 176.0 seconds, 104.8 seconds from Rob, 71.2 seconds from Lidewij. Figure 4.3 shows a chronogram of the distribution of the utterances and stressed syllables in the conversation. The conversation is split up in three parts to be able to display the figure on one page. As can be seen in the figure, the participants are sometimes speaking at the same time, a lot of times no-one is speaking, but most of the time just one person is speaking. In chapter 4.3.2 we will introduce the concepts of *floor management* and *turn taking* to help to explain these characteristics.

speech prosodic accents

The identified speech prosodic accents, or stressed syllables are marked with a highlighted region in figure 4.3. The speech prosodic accent fragments have a lighter color than the utterance fragments. In the data 109 speech prosodic accents are identified, 69 from Rob, 40 from Lidewij. The figure shows the utterance fragments in time as horizontal bars. The x-axis represents the time since the start of the conversation, the y-axis has two discrete points: L for the person sitting on the left of the video (Lidewij), R for the person sitting to the right (Rob). A filled bar, green for L, red for R, represents one utterance by either L or R. The conversation is split in three parts to be able to show the whole conversation on one page. As can be seen in the figure, longer periods of talk (subsequent utterance fragments spanning more than 5 seconds) by one person are varied with longer periods of talk by the other person. Shorter periods of talk (<5 seconds) occur during other person's talk, after a longer period of talk by the other, or after a moment without talk. Throughout the conversation there are periods without talk, sometimes these can be very long (>10 seconds), other times very short (<0.1 second).

gaze labels

The distribution of the labels AT, BLINK and AWAY over time is depicted in figure 4.4. The figure shows the gaze state over time for each of the participants as horizontal bars of different colors: AT fragments are lime colored, AWAY aqua and BLINK red. It can directly be seen that the gaze state durations of Rob (R) are mostly short compared to those of Lidewij (L). Also, Lidewij is blinking more, and has lengthy AT periods especially at the beginning of the conversation. In the third part of the figure there are long stretches of away for both participants, at this time Lidewij was writing down one of the questions, and they both look down at the paper.

Table 4.3 shows the total and per participant counts of the different gaze labels, plus the count of a group of labels referred to as AWAY. The AWAY label group contains all labels except AT and BLINK. The table shows that 75

Table 4.3: gaze and blink label counts

	at	*blink*	*R*	*L*	*U*	*UL*	*UR*	*D*	*DL*	*DR*	*C*	*away*
Lidewij	75	102	30	1	0	0	0	7	0	25	11	74
Rob	55	9	1	8	0	0	1	43	44	4	6	107

segments on the gaze tier of Lidewij were labeled AT and 74 were labeled AWAY, For Rob this is 55 AT and 107 AWAY. Lidewij blinks a lot (102 times) while Rob blinks only 9 times. The label counts in the AWAY group for Lidewij are highest with R and DR, while with Rob they are mostly D and DL. Interesting to note

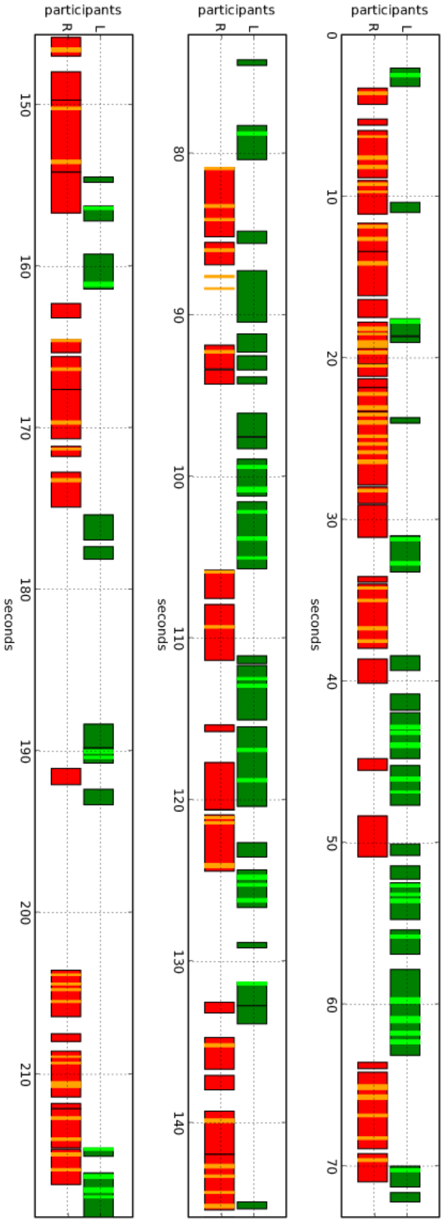


Figure 4.3: chronogram of utterances and stressed syllables

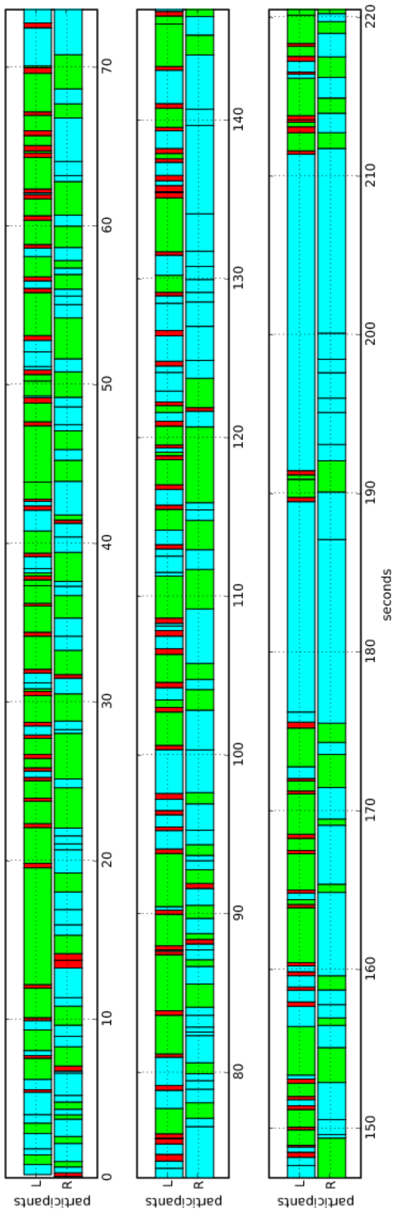


Figure 4.4: chronogram of eye gaze and blinking

is that both participants, *if* they are looking away, they tend to look away in the opposite direction as where the other person is sitting. What this means is that while Lidewij is sitting on the left (from the viewers point), thus Rob is sitting left of her (from Lidewij's point of view), she tends to gaze away from Rob in the opposite direction. For Rob this is the other way around, although the down-component is stronger in his case; the L label count is not that high with Rob.

These are however only the label *counts*, so it doesn't tell us about how much time is spent gazing AT or AWAY. The following two tables, 4.4 and 4.5, list respectively the total time per label and the percentage of time per label. The first is calculated by the sum of all segment durations having a specific label (label time), the latter by $\frac{label\,time}{total\,time} \cdot 100\%$ (label time percentage) . For completeness table 4.6 lists the average duration per label. From the label time percentages, in table 4.5, we can conclude that (time-wise) Lidewij is gazing at Rob more than the other way around: 56% of the time Lidewij is looking at Rob while Rob is looking 33% of the time to Lidewij, blinking time not included. The average duration per label, from table 4.6, for the AT label is 1.5 seconds for Lidewij and 1.3 for Rob. Interesting from this table is that the average duration Lidewij gazes down, is very high (5.4 seconds) compared to the other labels. If we look closely at figure 4.4 however, this can be explained. In the figure it can be seen that Lidewij's and Rob's gazing away time is for the most part situated in the period from time 176s to 212s. In this part of the conversation Lidewij is writing down one of the questions they came up with, during which both participants are mainly look down at the paper on the table.

Table 4.4: total gaze time per label (seconds)

	at	*R*	*L*	*D*	*DL*	*DR*	*C*	*away*
Lidewij	109.5	20.1	0.1	37.7	-	21.6	6.6	86.1
Rob	71.7	0.4	4.9	81.1	53.2	2.9	2.1	146.4

Table 4.5: percentages of total gaze time (%)

	at	*R*	*L*	*D*	*DL*	*DR*	*C*	*away*
Lidewij	56	10	0	19	-	11	3	44
Rob	33	0	2	37	24	1	1	67

Table 4.6: average gaze durations (seconds)

	at	*R*	*L*	*D*	*DL*	*DR*	*C*	*away*
Lidewij	1.5	0.7	-	5.4	-	0.9	0.6	1.2
Rob	1.3	-	0.6	1.9	1.2	0.7	0.4	1.4

So variations seem to occur between persons, and dialogue contexts. The relation of gaze with the turn-taking process and other contexts is further analyzed in section 4.6.

head movements

The data contains 326 head movements, 163 from Rob and also 163 from Lidewij. Table 4.7 shows the counts, table 4.8 the durations of the head movements per label and per rotation axis.

Table 4.7: head movement counts

	⌒x	*~x*	*/x*	*x*	*⌒y*	*~y*	*/y*	*y*	*⌒z*	*~z*	*/z*	*z*
Lidewij	30	3	57	90	4	1	49	54	2	0	17	19
Rob	17	8	49	74	6	4	66	76	1	0	12	13

Table 4.8: average duration of head movements

	⌒x	*~x*	*/x*	*⌒y*	*~y*	*/y*	*⌒z*	*~z*	*/z*	*total*
Lidewij	0.5	0.9	0.5	0.6	1.4	0.5	0.2	-	0.5	0.5
Rob	0.6	0.7	0.4	0.5	0.5	0.4	0.2	-	0.4	0.5

The tables show that the simple nod (⌒x), the vertical move (/x) and horizontal move (/y) are most common. Rob uses the nod less than Lidewij: 17 against 30. Furthermore, it can be noticed that movements around the z-axis (roll) are not that common (17 for Lidewij, 12 for Rob), but if they occur it is more likely a move in one direction (/z) than nod-like (⌒z, ~z). As can be seen in table 4.7 the nod with overshoot (~) labels are almost never used.

Since it was not always easy to choose the right label, it might be that the label set is not sufficient to capture the head movements. When segmenting the video data, it became clear that head movements can be complex sometimes, this means that a movement can be made up of smaller movement *parts* in different directions together forming a whole movement. So the choice was made to segment such complex movements into smaller components, and using the defined label set on the smaller segments. In the previous chapter, in subsection 3.4.2, a different annotation scheme was used for head movements, which took more components of the head movement trajectory into account. For example *modifiers* such as ARCH, JERK and REPEATED allowed in this scheme to capture the main trajectory of a movement, while also taking the small deviations into account. Also, in that annotation session a segment was allowed to contain *multiple* head movements. If we compare the average duration of a head movement segment as annotated here (see also table 4.8), with the *single* head movement segments as used in the previous chapter (table 3.2), we can see that the segments here are shorter: 0.5 seconds compared to 0.8. The following list summarizes these and other problems encountered while using the defined label set as described above:

- The data contains head movements that have a rotation over multiple axes at the same time; the label set does not contain e.g. xy movement combinations,
- movements can be a combination of smaller movements,
- a nod is sometimes annotated as two consecutive single movements (/),

- the difference between 'nod' and 'nod with overshoot' was not always clear while observing the movement,
- small movements are sometimes identified as movement, sometimes not.

Since the head movement tiers have non-contiguous segments, it's also interesting to see, like the utterances, what the distribution of the movements over time looks like. This is shown per axis in figure 4.5. The figure shows head movement fragments as horizontal bars with the different colors for the different rotation axes: aqua colors bars represent head rotations along the x-axis, lime along the y-axis, and red along the z-axis. What we can further point out looking is that there are periods with a density of head movements, and periods where head movements are rare. Interesting about this is that it appears that, when one participant moves his/her head a lot, the other is likely to do the same.

4.3 Dialogue acts, floor & mirroring

The former section showed us how the data was collected, which behaviors or aspects of behaviors were transcribed, and some basic counts and ratios of the transcriptions. An important property of the transcriptions made here, and probably transcriptions in general, is that they intend merely to *describe*, or *record*, what happened. Contrary to annotations on data that have a functional connotation, the transcriptions by itself lack semantic value. In other words where the transcriptions of the former section focus on the form, or observable properties, of behaviors, the annotations presented in this section are about how the behaviors function in a specific context. In this section three aspects of the conversation dealing with interactive processes during conversations are annotated. These aspects are:

- utterance dialogue acts
- floor ownership
- mirroring episodes

For the purpose of a detailed analysis of interactive processes, adding these aspects to our data will leverage the more detailed analyses of these processes performed later in this report in sections 4.5 and 4.6. So before diving into different theoretical views, and problems associated with more interpretative annotations, the annotation scheme's and a short description of their theoretical foundations are presented.

4.3.1 Utterance dialogue acts

Bunt [Harry Bunt, 1996, 1994] says about dialogue acts that they are:

> *functional units used by the speaker to change the context*

In this abstract notion of dialogue act, *functional units* are distinguished from the *context* in which they appear, and on which they operate. This means that these units have a meaning separate from the specific context in which they

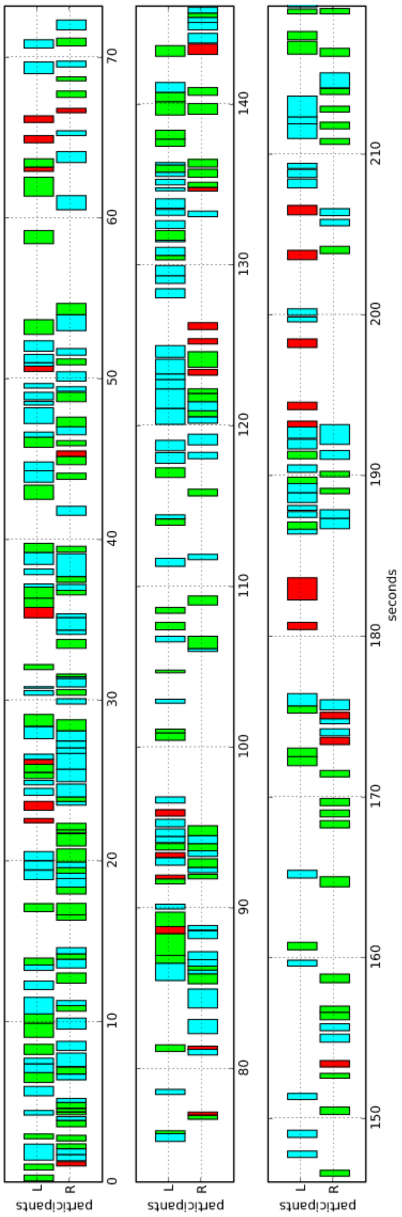


Figure 4.5: chronogram of head movements

are used. Also, this makes it possible to label these units using a annotation scheme such as DAMSL. DAMSL (Dialogue Act Markup in Several Layers) is a well known annotation scheme intended to give insight into the functional aspects of utterances. The nature of the conversations analyzed in this chapter seems to fit with the primary focus of DAMSL which are dyadic task-oriented dialogues. The difference between the conversations DAMSL focuses on and the conversation analyzed in this chapter, is that with DAMSL the task is to “collaborate to solve some problem”, whereas the task-oriented conversations in the GT2M data is more about brain-storming and reaching mutual agreement. Nonetheless, both types of conversation have a clear description of *task*, and for both, interaction between the participants is necessary to reach the goal of the conversation.

The DAMSL manual [James Allen and Mark Core, 1997] defines tags to be given to utterance units in an annotation in four layers, each layer composed of a set of tags listed here. The terms between brackets are utterance tag labels.

- Communicative status: In this layer a tag can optionally be given to utterances if they don’t contribute to the content of the conversation (abandoned, self-talk) or if the meaning of the utterance cannot be established (uninterpretable).

- Information level: Tags in this layer indicate if an utterance addresses the task (task, and task management) or not (communication-management, other).

- Forward looking function: Is about the effect of the utterance on the subsequent dialogue and interaction. There are eight aspects that can be codified in this layer:

  - statement: The purpose of the utterance is to make a claim about the world; something that can be true or false. Labels: (assert, reassert, other).
  - influencing-addressee-future-action: Purpose of utterance is to influence the hearers non-communicative action. (open-option, action directive).
  - (info-request): From the manual: “Utterances that introduce an obligation to provide an answer”.
  - committing-speaker-future-action: The speaker commits him/herself to doing something. Labels: (offer, commit).
  - conventional: Greetings and conventional introductions et cetera. Labels: (opening, closing),
  - explicit-performative: Not used in the annotation here.
  - exclamation: Not used in the annotations here.
  - other: Other forward-looking function.

- Backward looking function: Indicates the relation between the current utterance and the previous dialogue. There are four aspects:

  - agreement: The utterance's purpose is to affect what is agreed upon, mostly concerning the task. Labels: (accept, accept-part, maybe, reject-part, reject, hold).
  - understanding: Covers different aspects of understanding on what was uttered earlier in the dialogue. Labels: (signal-non-understanding, acknowledge, repeat-rephrase, completion, correct-misspeaking).
  - (answer): Usually follows after an info-request.
  - information-relation: not used here.

Except for the communicative status layer and some unused labels, these *utterance tags* are applied to the utterance annotations from section 4.2. The term *utterance tag* used in the DAMSL manual is a label ascribed to one or more *utterance units.* Since the forward and backward looking function allow for multiple tags per utterance multiple tiers per participant per layer are defined in ELAN. DAMSL allows, but does not enforce, that each utterance or several subsequent utterances can get a tag from each layer.

The information level layer tries to separate tasks with task management, therefore, as noted in the manual, the tasks have to be clearly specified. In the scenario of the task-oriented GT2M conversations the task ('doing the task') was to formulate three questions. This also includes:

- discussing about and agreeing on the questions to ask, and
- writing the questions down

Examples of the speech activities associated with this task are:

- proposing, formulating or reformulating a question
- judging these proposals and formulations

Examples of the task management ('talking about the task') process activities are:

- restating the task and agreeing on the interpretation of thereof,
- identifying, and agreeing on the type of questions to ask
- arguing why certain types of questions should or should not be asked, considering the task to be performed.

The fragment studied in the annotation session got 207 utterance tags, 88 in the information level layer, 76 in the forward looking function, 43 in the backward looking function. Table 4.9 lists the tag counts per layer and per participant. Tags with count zero are not listed. The table shows that on the information level layer most utterances are dedicated to task management. When looking at the forward looking function *info-request* and *action-directive* tags are rare. *Offer* and *commit* tags are common, as well as *assert* tags. The backward looking function has two dominant tags: *accept* and *acknowledge.* Furthermore it can be noticed that this is mostly due to tags with utterances from Lidewij. While the other layers have equally spread tag counts for both participants, Lidewij is clearly more active in using the backward looking function with utterances.

Table 4.9: utterance dialogue act tag counts

		counts		
layer	*tag*	*Lidewij*	*Rob*	*total*
information level	task	11	8	19
	task management	16	24	40
	communication management	11	9	20
	other	5	4	9
forward looking	assert	8	10	18
	reassert	1	-	1
	info-request	3	-	3
	open-option	9	5	14
	action-directive	3	-	3
	offer	11	10	21
	commit	7	5	14
	other	1	1	2
backward looking	accept	13	4	17
	accept-part	1	2	3
	reject	2	-	2
	maybe	-	1	1
	hold	1	-	1
	acknowledge	6	2	8
	repeat-rephrase	4	1	5
	correct-misspeaking	-	3	3
	completion	1	-	1
	answer	-	2	2

4.3.2 Floor ownership

The turn-taking process is a process in which, ideally, all participants in the conversation contribute to managing it. In this subsection first a simplified view on the turn taking process is adopted, in which the two participants are assumed to have the following activities:

- take turn
- give turn

This simplified turn-taking system has two rules:

- the take turn can be performed by one conversant if the directly after the other performs a give turn, and
- the give turn can only be performed by one conversant after a take turn by the same conversant.

After one conversant starts the conversation with a take turn, this system assumes both conversants keep to the rules. This system is adopted here as a first effort for specifying the turn-taking process, to apply it to some real conversation and see if it holds. To see if this system holds, and (more importantly) on which aspects it does not hold, we first try to identify fragments from the video that can be labeled with one of the two activities. In the annotations each participant has a tier with a controlled vocabulary holding two possible tag values: take, give. Figure 4.6 shows the utterance fragments combined with the turn signals marked by the light green and blue colored bars. Turn-taking signals annotated vary from blinks, head nods, manual gestures to speech stops. A specific behavior was labeled as turn-take or turn-give if the behavior appeared to function as such, but absence of behaviors was not considered a turn-signal. The figure shows the times in the conversation when a turn-taking signal was observed by the annotator, it also contains the utterance fragments, as shown earlier in this section, and floor ownership, which will be explained later in this section. The utterance fragments are the green and red bars closest to the middle of the y-axis. The turn-take and turn-give signals are the shorter horizontal bars above and below the utterance fragments, lime representing TAKE, blue GIVE. The floor ownership fragments are represented by the longer aqua colored bars at the top and bottom.

As expected, the adopted turn-taking system presented above does not hold given the annotated tags. This is also reported by Kristinn R. Thórisson [2002]:

> From the discussion so far it is clear that a step-lock "transmitter/receiver" model will not be sufficient when imparting multimodal interaction to the computer. Back-channel feedback, interruptions, real-time construction, unforeseen events all hint at a much more complex, dynamic system in which multiple states and events serve to provide a rich context for the participants' mental processing.

The following list shows the cases where the presented system does not hold given the annotations:

1. take signals at the same time (see around t=6s and t=91s)
2. subsequent take signals by the same person (t=6s, t=96s, t=115s)
3. give not followed by a take (t=45s, t=127s)
4. take without a give and without this being an interruption (t=51s, t=115s, t=187s)

As the above suggests, another way of looking at the turn-taking process is needed, to explain what happens when people exchange speaking turns, therefore the notion of floor ownership is introduced here. The reason to choose the term floor ownership here is that we want to make a clear distinction between the process of using signals of turn-taking/giving, and the notion that, when watching the video data, it seems most of the time it is clear to the conversants who is has the turn to talk or is going to talk soon. So the way floor ownership is defined here is that one conversant has it if it clear who's turn it is to speak. To illustrate: if we look at figure 4.6, we can see that if one person is speaking the other person is not. It looks like, in some way, a mutual agreement exists on who is 'allowed to talk'. This doesn't mean though that conversants are constantly aware and conscience about these agreements. Floor ownership is thus an important characteristic for determining who's turn it is to talk or going to talk.

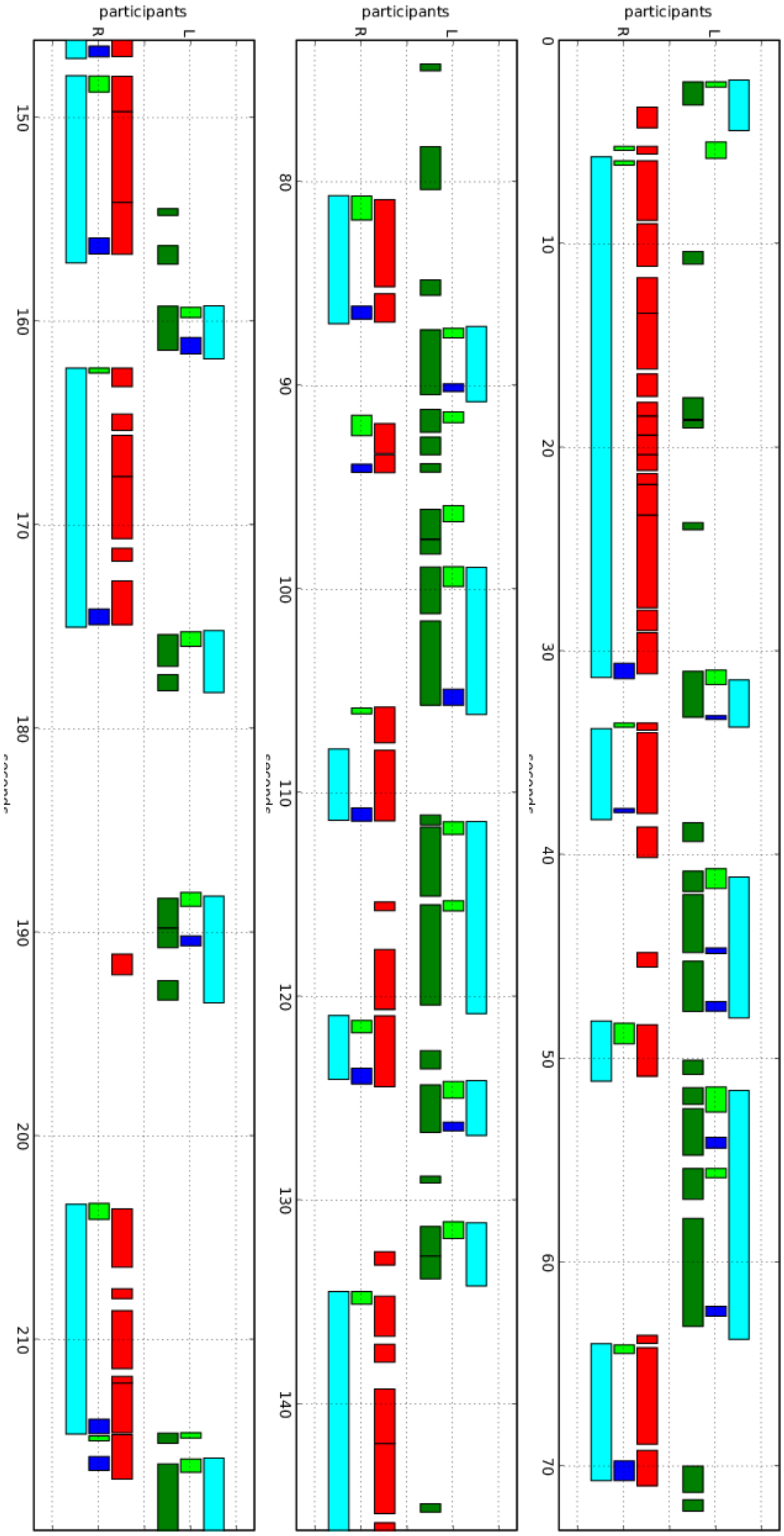


Figure 4.6: turn taking, giving and floor ownership with utterances

The process of establishing this agreement (floor management) is rarely explicitly expressed through utterances such as “you may speak now”, especially in informal conversations. More often, communication about floor management is more subtle, and coordinated by using non-verbal behaviors such as gaze. As Goodwin [Charles Goodwin, 1981] points out after investigating behaviors at turn beginnings:

> It has been found that the gaze of both parties is a relevant feature of face-to-face conversation and that the participants have access to, and make use of, systematic procedures for achieving appropriate states of mutual gaze.

The point Goodwin makes here directs us to think that mutual gaze plays a role in floor management. This also aligns with our earlier assumption that turns can be exchanged using non-verbal turn-signs.

Note that a problem with using the notion of floor ownership and floor management, is that both participants do not necessarily share the same view on who has the floor; a mutual agreement is not always the case. For the observer/annotator this can also be a problem, especially in cases where simultaneous talk or long pauses occur. In figure 4.6 the time periods in which we believe that person L or R has the floor are marked respectively with the bars at the top and bottom. The figure shows that floor ownership is annotated here as an exclusive property, in other words only one person has the floor at the time, consistent with the definition of floor ownership described above. Furthermore is can be seen that sometimes the change of ownership is quick, other times it takes longer. Also there are periods when nobody has the floor. This can mean two things: (1) simultaneous talk is going on, or (2) nobody is talking for a longer period of time, nor is there anybody expected to take the floor soon. As mentioned, in chapter 4.6 we are looking into floor change (floor ownership change) and associated behaviors in more detail.

4.3.3 Mirroring episodes

Researchers in different fields studying conversational behavior have reported that mimicry or mirroring of several behaviors commonly occurs at various levels during conversation (e.g. [Lakin, 2003, Dirk Heylen, 2005b, Jonathan Gratch et al., 2006]).Here we try to limit ourselves to annotating two types of mirroring behavior:

1. exact synchrony of body, eye or head movements
2. behaviors of one person repeated by the other shortly after the behavior started (within one second).

The reason to define 2 different types of mirroring, is that type 2 has two properties of interest that type 1 hasn’t:

- delay time between mirroring and
- initiator, the person making the first move.

Also other types could be annotated, e.g. types with mirroring delays of more than one second, but focus here is on short time range mirroring behavior.

The three and a half minute contains 11 mirroring episodes of type 1, 10 of which contain head movement mirroring, 3 gaze and 8 body movements. The duration of these episodes ranges from 0.9 to 5.9 seconds. 10 episodes of type 2 were identified, 8 had gaze, 5 head movement. Rob started 3, Lidewij 7.

4.4 Finding patterns in the data

One way of finding regularities in conversational behavior is to use annotations from a face-to-face conversation, and look at *co-occurring behaviors.*The following section presents a way to search the data and get insights into Listener behavior in specific contexts and how it relates to Speaker behavior.

4.4.1 Regions of interest

To find patterns of Listener and Speaker behavior in specific contexts, the following strategy is followed:

1. define different contexts of concern, and give each context a category label (*cat*). Examples of category labels are: smooth floor transition, unsmooth floor transition, feedback, et cetera.

2. for each category label, define subcategory labels (*subcat*). For example the smooth floor transition category might have two subcategories: transition from one person to the other, and from the other to the first.

3. for each category, select time points (*tp*) in the data that belong to that category, and assign the appropriate subcategory label.

4. Now if we want to know what happens, during, before or after a certain point in time, the region of interest for each of the time points can be defined by the parameters: t_{before} and t_{after}.

The above steps produces the regions of interest, defined by the parameters:

1. a list of time points *tp*, having two labels: *cat* and *subcat*. This list will be referred to as *pois*

2. t_{before}: the start times of the regions of interest are: $tp - t_{before}$.

3. t_{after}: the end times of the regions of interest are: $tp + t_{after}$.

4.4.2 Counting co-occurrences

Since our interest lies in finding relations between Listener behavior and Speaker behavior in certain contexts, we can look at behaviors of the Speaker co-occurring with behaviors of the Listener. This means that during a fragment (region of interest) a combination of two specific behaviors are of interest: a behavior performed by the Speaker, and a behavior performed by the Listener. Before counting the number of co-occurrences, we can define a list of behavior combinations (*coocc*) that we want to look into. For example if we had annotated some gaze behavior we can make a list of four behavior combinations (*bc*): {AWAY, AWAY}, {AWAY, AT}, {AT, AWAY}, {AT, AT}. A behavior combination pair consists of two behavior labels, one from the person sitting to the left, the other from the person at the right. This example assumes that the annotation data contains the AWAY and AT labels for each of the two persons.

To count the number of co-occurrences in the regions of interest a function is devised that takes the following parameters:

1. $rois$: regions of interest defined by $pois$, t_{before} and t_{after}
2. *coocc*: list of co-occurrences to be counted
3. sr: sample rate in Hz.

The function *countCoOcc* works as follows:

1. for each roi in $rois$:
2. take samples (s) from the roi at sample rate sr.
3. for each behavior combination (*bc*) in *coocc*: compare it to s,
4. if s matches a *bc*, add one to the count of *cat*, *subcat* and the *cat.subcat* combination.

4.4.3 Using co-occurrence counts to find patterns

Applying the *countCoOcc* function to the data, gives three lists containing the co-occurrence counts of category, subcategory labels, and combinations thereof. To find patterns it might be useful to construct different versions of the count lists by varying over the parameters t_{before}, t_{after}, *sr* and *coocc*. These lists then can be further analyzed along different lines by comparing the counts of one list to that of another constructed with different parameters. Another way is to compare different categories or subcategories constructed with the same parameters. So the count lists can be both used as a means to find parameters that give useful results, as well as to find out if specific behavior combinations co-occur in a specific context.

4.5 Counting co-occurrences

Following the recipe explained above (section 4.4), the annotated data is scanned for co-occurrences. The first task is to define the points of interest, in other words the time-line of the conversation is labeled with *categories* and *subcategories*. As mentioned at the start of this chapter our interest lies in finding patterns for three interactive processes: turn-taking, feedback and mirroring. For each of these processes one or more category and subcategory labels can be defined. The main question to be answered is: What co-occurring behaviors distinguish one context (defined by category and subcategory) from another? Section 4.6 answers this question for each of the three main contexts: turn-taking, feedback and mirroring. First however, agreement on the categories and subcategories for the contexts has to be established.

4.5.1 Turn taking

As explained in section 4.3.2, the turn-taking process is annotated in terms of *floor ownership*. In our situation the floor can be owned by one person, the other, or no-one. During the conversation ownership of the floor changes. If we look again at figure 4.6 and try to find episodes of floor change in the data, we can see that floor change is maybe too loosely formulated a term. Looking at floor change as a state transition from the one state (e.g. *ownerIsLidewij*) to the other (e.g. *ownerIsRob*), we define three types of floor transitions:

1. Smooth floor transition: the time between floor owned by one person followed by the other, is less than or equal to one second.

2. Floor to open: nobody owns the floor anymore for at least one second.

3. Other: floor ownership changes but none of the two types defined above apply.

These types of floor transitions add three category labels to our counting procedure: SMOOTH, FLOOR-TO-OPEN and OTHER. Furthermore the following subcategories are defined:

Table 4.10: floor subcategories

label	*category*	*transition*
LR	smooth, other	left to right
RL	smooth, other	right to left
Lo	floor-to-open	left to no-one
Ro	floor-to-open	right to no-one
oL	other	no-one to left
oR	other	no-one to right

The time points of interest (*tp*) for the floor categories are the end times of the floor ownership episode from the one who loses floor ownership, or the start time of the one who gains ownership (in case of *oL* and *oR*).

4.5.2 Feedback

Although feedback or back-channeling can manifest itself in many different ways, we can only use the data we have. Given the annotations of utterance dialogue acts it seems appropriate to use the utterance fragments containing a *backward looking function* to get insight into the behaviors associated with the feedback process. Note that although feedback can also be expressed non-verbally or even by absence of specific behavior, we only use the available data collected as described in section 4.3.1. We add FEEDBACK to our category labels. Table 4.11 lists the subcategory labels used in the FEEDBACK category: The ACCEPT and ACKNOWLEDGE functions are used here because they are the most popular backward looking functions in our data, see also table 4.9. The time point of interest (*tp*) for feedback is the start time of the utterance labeled with a backward looking function.

Table 4.11: feedback subcategory labels

label	*person*	*backward looking function*
Laccept	left	accept
Raccept	right	accept
Lacknow	left	acknowledge
Racknow	right	acknowledge
Lackacc	left	accept, acknowledge
Rackacc	right	accept, acknowledge
Laccpar	left	accept-part
Raccpar	right	accept-part

4.5.3 Mirroring

As described in section 4.3.3 we distinguish two types of mirroring. These types form our category labels: MIRRORING1 and MIRRORING2. Subcategories for type 2 have an initiator: L or R. However, it might also be interesting to see how a mirroring episode starts or ends, and since the episodes can be long we cross product the L and R subcategories with *start* and *end*. This gives the subcategories listed in table 4.12. The time points of interest (*tp*) for mirroring are both the start time and the end time of the mirroring episodes.

Table 4.12: mirroring subcategories

label	*initiator*	*category*
Lstart	left	mirroring2
Rstart	right	mirroring2
Lend	left	mirroring2
Rend	right	mirroring2
start	none	mirroring1
end	none	mirroring1

4.5.4 categories and subcategories in the data

Figure 4.7 shows fragments with floor ownership, feedback, mirroring and utterances as well as the category and subcategory labels. It shows which time periods in the conversation a label was set on the annotation tier. Table 4.13 lists the counts for the categories and subcategories. The figure and the table shown here mainly function as a reference for the analyses in the remainder of the chapter. When looking at the figure one can get an idea of what happened in the actual conversation, without having to watch the video.

4.5.5 varying parameters

Now the points of interest are defined, we still have three parameters left: *sample rate*, *time before* and *time after*. The outcome of using the *countCoOcc* function on the data with varying values for the parameters helps in defining real-time computational models for conversational agents, because, since an artificial conversational agent makes decisions about actions to take based on

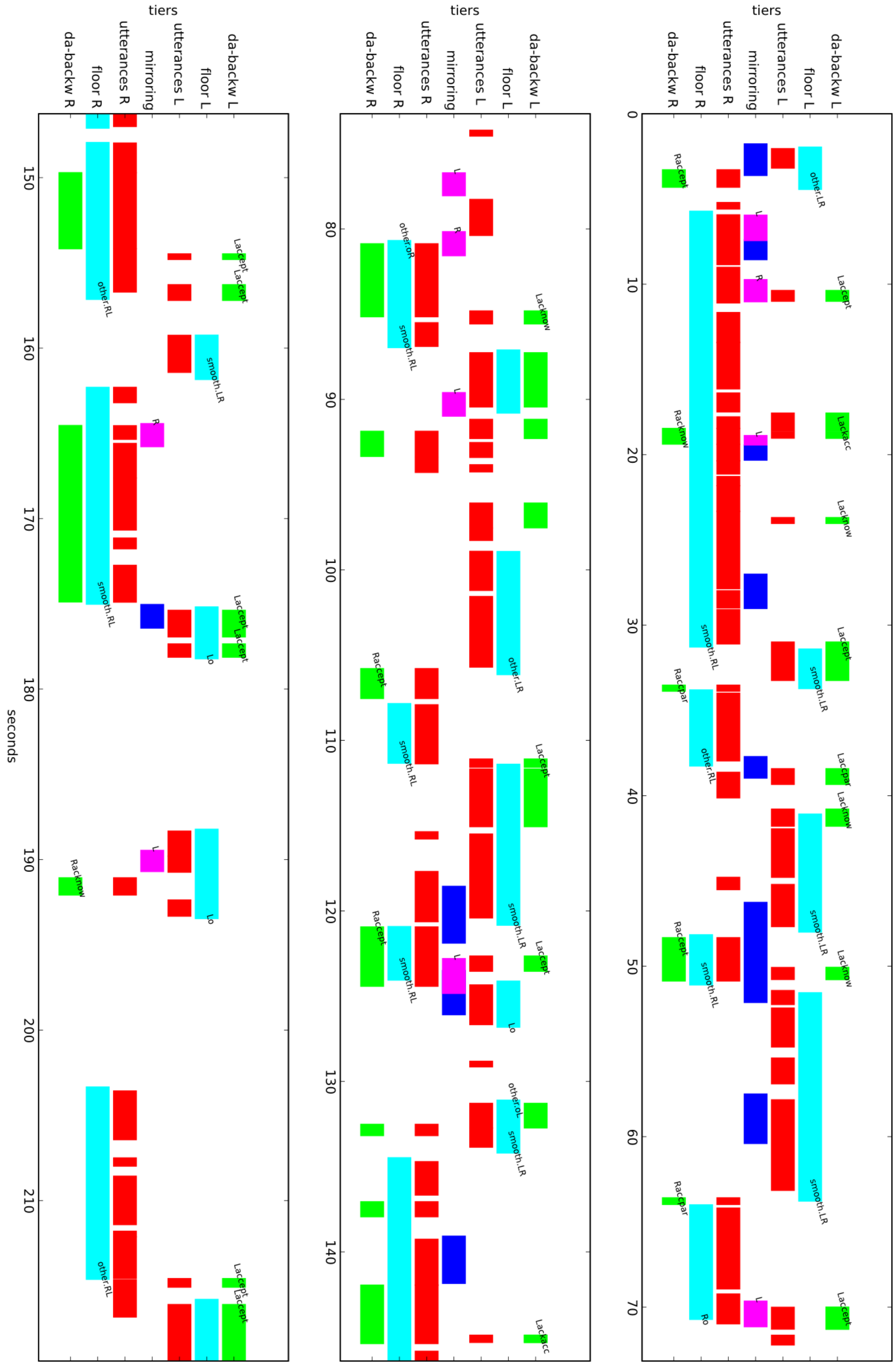


Figure 4.7: category and subcategories in the data

Table 4.13: category and subcategory label counts

category	*count*
smooth	11
floor-to-open	4
other	8
feedback	26
mirroring1	22
mirroring2	20

subcat	*count*	*subcat*	*count*	*subcat*	*count*
LR	8	Laccept	11	Lstart	7
RL	8	Raccept	4	Rstart	3
Lo	4	Lacknow	4	Lend	7
Ro	1	Racknow	2	Rend	3
oL	1	Lackacc	2	start	10
oR	1	Rackacc	0	end	10
		Laccpar	1		
		Raccpar	2		

provided information, it is crucial to identify which parameters (such as *sample rate*) are of importance and how these parameters influence measurements.

sample rate

The rate at which certain information is gathered (sampled) and processed may have an impact on real-time performance of the agent. Also the sample rate will pose certain limitations on the time within which an agent can react. For example if an agents needs to react to certain behavior of another agent, but it samples its environment at a rate of 1Hz, it may be (worst-case) that the time to react for the agent is about one second after the action from the other agent, which may be too long. As Thórisson points out in [Kristinn R. Thórisson, 2002] the time it takes to make a lot of decisions can be as less as 460 milliseconds. This can also be seen in the data presented here. For example there are 11 SMOOTH floor transitions out of 23, which could mean that decisions about floor management are taken within one second.

To find out what impact the sample rate parameter has on the counts the data with the following parameters is explored:

- two contexts: the whole conversation (ALL) and smooth turn transition (SMOOTH),
- co-occurring gaze behaviors: {AWAY, AWAY}, {AWAY, AT}, {AT, AWAY}, {AT, AT},
- constant time before and time after: both 1 second,
- varying sample rate: 1, 2 and 4 Hz.

The results are listed in table 4.14. Comparisons of this kind of multidimensional data, as presented in the table can be performed in many ways. In table 4.15, a column with percentages over the whole category (% cat) per behavior combination, and a column (% sr) ranging over the sample rate and category is listed. This enables comparison of behavior combination counts over the whole category, as well as per sample rate.

The counts of co-occurrences in the whole conversation (ALL) are predictable over the different sample rates: if the sample rates double, the counts also double more-or-less. The reason that the doubling of the sample rate does not always mean a doubling in count is that the counting function samples the data by

Table 4.14: co-occurring gaze behavior counts with varying sample rates

cat=*all*,sr=1		**count**	**cat=*all*, sr=2**		**count**	**cat=*all*,sr=4**		**count**
away	away	82	away	away	171	away	away	334
away	at	27	away	at	51	away	at	111
at	away	63	at	away	121	at	away	249
at	at	43	at	at	88	at	at	169
cat=*smooth*,sr=1		**count**	**cat=*smooth*,sr=2**		**count**	**cat=*smooth*,sr=4**		**count**
away	away	4	away	away	10	away	away	24
away	at	4	away	at	4	away	at	10
at	away	8	at	away	19	at	away	40
at	at	8	at	at	15	at	at	22

Table 4.15: co-occurring gaze behavior percentages with varying sample rates.

cat:*all*, sr:1		**% cat**	**% sr**	**cat:*all*, sr:2**		**% cat**	**% sr**	**cat:*all*, sr:4**		**% cat**	**% sr**
away	away	5.4	38.1	away	away	11.3	39.7	away	away	22.1	38.7
away	at	1.8	12.6	away	at	3.4	11.8	away	at	7.4	12.9
at	away	4.2	29.3	at	away	8.0	28.1	at	away	16.5	28.9
at	at	2.8	20.0	at	at	5.8	20.4	at	at	11.2	19.6
cat:*smooth*, sr:1		**% cat**	**% sr**	**cat:*smooth*, sr:2**		**% cat**	**% sr**	**cat:*smooth*, sr:4**		**% cat**	**% sr**
away	away	2.4	16.7	away	away	6.0	20.8	away	away	14.3	25.0
away	at	2.4	16.7	away	at	2.4	8.3	away	at	6.0	10.4
at	away	4.8	33.3	at	away	11.3	39.6	at	away	23.8	41.7
at	at	4.8	33.3	at	at	8.9	31.3	at	at	13.1	22.9

asking *if* a behavior combination is *present* at the time of sampling. Which could also mean that the sample *just* misses the behavior labels. In the SMOOTH category however, the doubling effect does not always occur. In table 4.15 it can be seen that SMOOTH {AWAY, AWAY} at sample rate of 1Hz has 16.7% of the total count in this category, sample rate combination, while at sample rate 4Hz this is 25.0%: an unexpected increase compared to the percentages in the ALL category: 38.1% to 38.7%. At the same time {AT, AT} goes from 33.3% to 22.9% in the SMOOTH category, compared to 20.0% to 19.6% in the ALL category. One way to translate this difference, caused by varying over the sample rate, is that:

> *During smooth floor transitions a gazing at by one person follows a gazing at by the other person more often within 0.5 seconds than after 0.5 seconds, relative to other behaviors following each other.*
>
> *Also gaze away behavior is followed by a gaze away more often after 0.5 seconds than within 0.25 seconds.*

What the above shows is that varying over the sample rate parameter can give insight into *timing properties* of certain behavior combinations.

time parameters

Now let's see how varying over the time parameters, *time-before* and *time-after*, influences the counts. Again the gaze behavior combinations are studied in relation to smooth turn transitions. This time however subcategories LR and RL are added to be able to distinguish Listener and Speaker from each other. *Sample rate* is kept constant at 4Hz, *time-before* and *time-after* are varied, so that it can be seen what happens just before a floor transition, and what happens after. Tables 4.16 and 4.17 list the result. Table 4.16 shows the counts for the floor transition from Lidewij to Rob. When comparing what happens only before such a transition (t_{after}=0.0) with what happens before and just after (t_{after}=0.5), the main difference lies in the count increase from the {AT, AWAY} co-occurrence. This means that in a smooth turn transition, in the time period just after Lidewij turns over the floor to Rob, in 10 samples out of 12 Rob is looking away while Lidewij is gazing at Rob. Furthermore, notice that the higher {AT, AT} counts completely disappear when looking at what happens after the turn transition (t_{before}=0.0, t_{after}=1.5). This result is not found in table 4.17, where only the floor transitions are counted from Rob to Lidewij. Here comparing before with after shows mainly a shift from {AT, AT} behaviors to {AWAY, AWAY}. But it does mean that it is quite common in our data that the person taking the floor tends to gaze more at the other person before a smooth turn transition than after.

Table 4.16: sr=4, cat=smooth, subcat=LR

t_{before}=1.5, t_{after}=0.0		**count**	**t_{before}=0.0, t_{after}=1.5**		**count**
away	away	2	away	away	10
away	at	4	away	at	2
at	away	17	at	away	24
at	at	13	at	at	0
t_{before}=1.5, t_{after}=0.5		**count**	**t_{before}=0.5, t_{after}=1.5**		**count**
away	away	4	away	away	11
away	at	4	away	at	2
at	away	27	at	away	32
at	at	13	at	at	3

Table 4.17: sr=4, cat=smooth, subcat=RL

t_{before}=1.5, t_{after}=0.0		**count**	**t_{before}=0.0, t_{after}=1.5**		**count**
away	away	3	away	away	21
away	at	6	away	at	8
at	away	8	at	away	2
at	at	19	at	at	5
t_{before}=1.5, t_{after}=0.5		**count**	**t_{before}=0.5, t_{after}=1.5**		**count**
away	away	9	away	away	23
away	at	9	away	at	10
at	away	8	at	away	5
at	at	22	at	at	10

So given this data, it seems that varying over the time parameters can give us insight into what *behavior transitions* take place in a specific context, and thus which behaviors are common before and after a change of dialog state.

4.6 Behaviors in context

Now we know roughly how varying over the parameters influences the results, we take a look at the interactive processes and try to find out which behavior patterns can be associated with the specific contexts in our data. First, in subsection 4.6.1, behaviors in the context of smooth floor transitions are compared to the same behaviors in other contexts. Then in subsection 4.6.2, we zoom in on head movements in the context of back-channeling. And later in subsection 4.6.3, shakes and nods are highlighted and considered in the different contexts. The last subsection (4.6.4) shows which behaviors can be associated to mirroring in our data.

4.6.1 Smooth floor transitions vs other dialogue contexts

Some researchers have noted that people make predictions about when a floor transition might occur. This leads us to believe there are specific behaviors that mark the upcoming transition. In [Kristinn R. Thórisson, 2002] Thórisson notes that *clues* that help conversants in managing the turn-taking process, are not only present in the audio signal, but other behaviors could be used as signals as well:

> If we assume that a listener is continuously looking for clues to classify each utterance we might conclude that the only features that matter are present in the stream of the audio signal. But this would be a mistake: Anyone who ignored all but the audio signal in a multimodal interaction would be throwing away a wealth of information that can be gleaned from the utterer's behavior pertaining to both the content and the process of the dialogue. We can be pretty certain that the 'evidence' people use to classify turn segments includes a number of sources, all the way from gaze to facial gesture to body stance (Taylor & Cameron 1987, Goodwin 1981).

Following this line of reasoning we have to assume that behaviors happening *before* a floor transition hold clues about *if* and *what kind* of floor transition will come up.

To find out if a smooth floor transition can be distinguished from other fragments in the conversation just by looking at gaze and head movement behavior of the participants, a look is taken at behaviors of one person that co-occur with behaviors of the other. The co-occurring behaviors in the following context categories and subcategories are explored:

- SMOOTH.LR, SMOOTH.RL,
- OTHER.LR, OTHER.RL,
- FLOOR-TO-OPEN.LO, FLOOR-TO-OPEN.RO, and
- FEEDBACK.L*, FEEDBACK.R*

Only co-occurring behaviors of gaze and head movement will be counted. The counts are performed with the *countCoOcc* function, as described in section 4.4, using a sample rate of 4Hz and a t_{before} of 1.5 second and a t_{after} of

0.0 second. Tables 4.18, 4.19, 4.20 and 4.21 list the results sorted by count. The fourth column (%) shows the percentage per sample, n gives the number of fragments in the category, subcategory combination, so the values in this column are calculated by: $\frac{count}{n \cdot 1.5 \cdot 4} \cdot 100\%$. These values tell something about how long a certain behavior combination was active on average within this *cat, subcat* combination. The L* and R* subcategories in table 4.21 are the sum of all feedback subcategories respectively with the L and R prefixes. The tables are sorted by the *subcat* column, this means that the first row contains the co-occurring behaviors of having the highest count in the data, while the last row has the lowest count.

Table 4.18: co-occurrences in the SMOOTH category

subcat=LR		**count**	**% (n=6)**	**subcat=RL**		**count**	**% (n=6)**
at	away	17	47	at	at	19	53
at	at	13	36	at	move	13	36
move	away	11	31	move	at	9	25
move	move	7	19	move	move	8	22
move	at	5	14	at	away	8	22
away	at	4	11	move	away	7	19
away	move	2	6	away	at	6	17
away	away	2	6	away	move	4	11
				away	away	3	8

Table 4.19: co-occurrences in the OTHER category

subcat=LR		**count**	**% (n=2)**	**subcat=RL**		**count**	**% (n=3)**
at	at	5	42	at	move	9	50
away	away	4	33	at	away	9	50
away	move	3	25	move	away	5	28
at	away	2	17	away	move	4	22
move	away	1	8	move	move	3	17
move	at	1	8	away	away	3	17
away	at	1	8	away	at	3	17
				at	at	3	17
				move	at	1	6

Table 4.20: co-occurrences in the FLOOR-TO-OPEN category

subcat=Lo		**count**	**% (n=3)**	**subcat=Ro**		**count**	**% (n=1)**
away	away	16	89	at	at	4	67
away	move	8	44	move	at	3	50
move	move	4	22	away	at	2	33
move	away	3	17	move	move	1	17
move	at	2	11	at	move	1	17
away	at	2	11				

Table 4.21: co-occurrences in the FEEDBACK category

subcat=R*		count	% (n=8)	subcat=L*		count	% (n=18)
at	at	19	40	at	away	39	36
at	away	15	31	at	at	39	36
move	away	13	27	at	move	33	31
at	move	11	23	move	away	22	20
move	at	10	21	move	move	21	19
move	move	10	21	away	move	20	19
away	at	7	15	away	away	19	18
away	away	7	15	move	at	12	11
away	move	6	13	away	at	11	10

When comparing the tables with each other, one can see that the top co-occurrences (having the highest count) are quite different for each table. For example the first two co-occurrences in the SMOOTH.LR category, subcategory combination are {AT, AWAY} and {AT, AT}, while in FLOOR-TO-OPEN.LO {AWAY, AWAY} and {AWAY, MOVE} have the highest counts. This means that having the same category does not at all mean that the percentages of co-occurring behaviors between subcategories are more-or-less the same.

Another difference noticeable from these table is that it makes a difference, for example, if the floor goes from Rob to Lidewij or the other way around. At the extremes this could be totally coincidental, because of sparse data or, at the other side of the spectrum, it could be that only the combination of Rob and Lidewij's personalities, roles or idiosyncratic way of communicating are accountable, also other aspects like the specific dialogue context, conventions or protocols play a role. In any case, because of this, here mostly comparisons are made between the results of different categories having the same or comparable subcategories.

Comparing table 4.18 with 4.21 shows that the top behavior combinations are more-or-less the same. For the LR and L* subcategories the top 2 co-occurrences are exactly the same: {AT, AWAY} and {AT, AT} in *both tables*. Looking at the RL and R* subcategories in these tables, one can see that they share the highest co-occurring behavior {AT, AT} and {AT, AWAY} and {AT, MOVE} are also high: position 2 and 4 and 4 and 2 respectively.

This last observation is not that surprising, since a lot of times in this data a smooth floor transition is followed by a utterance labeled as a backward looking function. This can be verified by looking at figure 4.7. Therefore, a new category is added containing *feedback* fragments that occur without the presence of a floor transition. This category is called BACKCHANNEL, the co-occurrence counts and percentages are listed in table 4.22. The BACKCHANNEL category clearly is more distinctive from the SMOOTH category than the FEEDBACK category. The {AT, AT} combination for example is much less prominent in the BACKCHANNEL category, with 25% and 29%, compared to the SMOOTH category, 36% and 53%. Head movement co-occurrences, however, seem to be *more* common, especially in the LR and R* subcategories: 33% and 27% in the FEEDBACK category against 19% and 22% in the SMOOTH category. Summarizing these last two features of the data, one could suggest that:

Table 4.22: co-occurrences in the BACKCHANNEL category

subcat=R*		count	% (n=2)	subcat=L*		count	% (n=8)
move	away	5	42	at	away	17	35
move	move	4	33	away	move	14	29
away	move	4	33	at	at	14	29
away	away	4	33	move	move	13	27
move	at	3	25	at	move	13	27
at	away	3	25	move	away	12	25
at	at	3	25	away	away	9	19
away	at	2	17	away	at	8	17
at	move	1	8	move	at	7	15

> *Before a smooth floor transition conversants gaze at each other longer than in the context of a back-channel relative to other behavior co-occurrences.*

Tables 4.18 and 4.19 also have some striking differences. If we look at Lidewij in the role of the Listener (subcategories RL) the SMOOTH floor transitions have on average 53% {AT, AT} co-occurrences in them, while in the OTHER category this is only 17%. So:

> *When Lidewij is listening, she and Rob gaze at each other much longer before a smooth floor transition than before a less smooth transition.*

In the LR subcategories, where Rob is the listener, {AWAY, AWAY} has a low count in the SMOOTH category (6%), while in the OTHER category this is 33%. So:

> *When Rob is listening the period before a floor transition takes place, features almost no gazing away from each other when the transition is smooth, while with a less smooth transition gazing away is not uncommon.*

4.6.2 Head movements in the context of Lidewij giving a back-channel signal

The tables also list head movement behaviors. Eye gaze and head movements are related. For example going from looking at one point in space to the other sometimes needs head rotations, because the eye rotation alone is not sufficient to get the other point in sight. Gaze at or away from one participant can both be accompanied with a head move from the same person. Looking at the co-occurrence counts in tables 4.18 through 4.22, it can be noted that co-occurrence behaviors containing a move are present, but not always in top. This raises the question of how much gaze changes are accompanied by head movements. To get a rough estimate of this, fragment *counts* are used. Rob has 162 gaze AT and AWAY fragments and 163 MOVE fragments this makes the ratio *gaze/move* 1.0. For Lidewij this is 149/163 = 0.9. These ratios don't necessarily mean that approximately every move is accompanied by a gaze change, but it *does* show that chances of finding a move in a sample are as high as finding a *gaze change*.

A big difference between gaze behavior and head movement behavior is that, in our data, gaze behavior is always present (whether it be AT, AWAY or BLINK), but movements are fragmented, in other words *gaze* is counted and not *gaze change*. So the total time of the annotated data is equal to the total time of the gaze behavior of one participant, while the total time of all *move* fragments is much less. Rob spends 75 seconds on movements, Lidewij 83 seconds, which is respectively $\frac{75}{218} \cdot 100\% = 34\%$ and $\frac{83}{196} \cdot 100\% = 42\%$ compared to the time spent on gaze at and away behaviors. The effect of this on the co-occurrence counts is that the chance to find a gaze at or away in a sample is a priori more likely than to find a move. One way to re-interpret the results found in tables 4.18 through 4.22 is thus to compensate for a priori likelihood. This means that, if we consider the percentages listed in these tables as if they represent the likelihood of a specific co-occurrence, the calculation of this likelihood must include the chance of finding a behavior combination in all of the data.

Here, one example is shown of how to compensate for likelihood in the context of Lidewij giving a back-channel signal, by looking at the likelihood of a co-occurrence in this context given the likelihood of the co-occurrence in all the data. The context category is BACKCHANNEL subcategory is L*. It has to be noted that although we use the terms chance and likelihood, no real statistical evidence is presented here, it is merely shown how to deal with this type of data, without taking reliability and side effects into account. The ALL category contains one fragment spanning the whole conversation. Co-occurrence counts found in the ALL category can therefore be used to estimate the a priori 'chance' a co-occurrence can be found in a sample. Using Bayes rule we can also calculate the chance a sample belongs to this category, subcategory combination, as listed in table 4.23.

Table 4.23: estimating back-channel chances given a co-occurrence

x		**cat=all**		**y=(cat=bc, subcat=L*)**		
L	**R**	**count**	**P(x) (n=863)**	**count**	**P(x|y) (n=48)**	**P(y|x)**
at	away	249	0.29	17	0.35	0.067
away	move	122	0.14	14	0.29	0.115
at	at	169	0.20	14	0.29	0.038
move	move	143	0.17	13	0.27	0.088
at	move	186	0.22	13	0.27	0.068
move	away	207	0.23	12	0.25	0.060
away	away	334	0.39	9	0.19	0.027
away	at	111	0.13	8	0.17	0.073
move	at	117	0.14	7	0.15	0.060

In table 4.23 *n* represents the number of samples. The P(y|x) column in the table gives us some idea what happens in the 1.5 seconds before Lidewij is giving a back-channel signal compared to what Lidewij and Rob normally do. Now the {AWAY, MOVE} co-occurrence has the highest score (0.115), {AWAY,AWAY} the lowest (0.027). This could be translated to e.g.:

When Lidewij is looking away and Rob is moving his head, chances *that Lidewij is going to use a back-channel signal is four times higher than when they both gaze away.*

4.6.3 Shakes and nods

Until now the analysis of co-occurrences did not differentiate between the types of head movements made. Researchers such as Kendon have confirmed that head movements, shakes in specific, can bear special meaning; meaning beyond "the kinesic equivalent of a unit of verbal expression". The most basic examples for these 'special meanings' are that nods can mean 'yes' and shakes can mean 'no', but the functions of e.g. shakes can be more subtle and there is a lot more they can express than just 'no', as noted in [Adam Kendon, 2002].

It would be interesting to see how shakes and nods in our data connect with the different contexts defined; in what context are nods and shakes used? As can be seen in table 4.7, 9 types of movement are used in the annotations. Here we use the term *cyclic movement* to refer to head movements performed in a repeated fashion; mostly shakes or nods. These are all the movement labels starting with ⌒ and ~ in scheme 4.2. Again, at a sample rate of 4Hz we can be reasonably certain that we don't miss any movements. Table 4.24 lists the counts for *occurrences* of cyclic movements. Occurrence counts are calculated in the same manner as co-occurrences, only each sample is inspected for a single behavior instead of behavior combinations (bc). This time the t_{before} and t_{after} parameters are both set to one second, which makes the sampling window for each category two seconds long, making eight samples per category fragment. The % column of tables 4.24 and 4.25 show how much time on average is spent on a shake or nod during category and subcategory fragments by Lidewij and Rob. In the normal case, the ALL category in table 4.24, both Rob and Lidewij are shaking or nodding 10% of the time. Looking only at the category

Table 4.24: occurrences of cyclic head movements per category

category	*total*	*L*	*%*	*R*	*%*
all	863	88	10	83	10
smooth	96	18	19	14	15
other	64	2	3	7	11
floor-to-open	32	4	13	4	13
feedback	208	29	14	32	15
backchannel	80	16	20	13	16

Table 4.25: occurrences of cyclic head movements per subcategory

subcat	*total*	*L*	*%*	*R*	*%*
LR	64	10	16	8	13
RL	64	8	13	13	20
Lo	32	2	6	4	13
Ro	8	2	25	0	0
L*	144	32	22	38	26
R*	64	13	20	7	11

contexts, Lidewij shakes or nods most during smooth floor transitions (19%) or backchannel fragments (20%) this also goes for Rob with 15% and 16%. Interesting is also the very low score in the other category, representing the floor transitions that are not smooth or floor-to-open, especially for Lidewij; in these 8 fragments she only nods or shakes for a total of 2 samples (3%), which is only 0.5 second.

Looking at the subcategory counts, in table 4.25, a lot of the percentages, like in the category fragments, are above normal. Also interesting is that, *if* Lidewij gets the floor or is giving feedback Rob is nodding or shaking double time or more.

4.6.4 Mirroring episodes

Section 4.3.3 explained how in some episodes during the studied conversation mirroring behavior fragments were annotated. The total time of mirroring episodes of type 1 is 26 seconds, which is approximately 12% of the whole conversation, for type 2 this is 14 seconds or 6%, a considerable amount of time. As mentioned these episodes are characterized by synchronized eye gaze and body and head movements. In type 1 episodes the same behaviors are performed at the same time, so one would expect that this can also be seen when counting the co-occurrences of head movements and gaze. Table 4.26 lists the co-occurrence counts and percentages of both types for synchronized behaviors as well as opposite gaze behavior combinations. A sample rate of 4Hz is taken a $t_{before}$of 0 seconds and a $t_{after}$of 1 second, so mainly the start of the mirroring episodes is inspected.

Table 4.26: behaviors after mirroring starts

		all	*mirr1.start*	*n=44*	*mirr2.Lstart*	*n=28*	*mirr2.Rstart*	*n=12*
L	*R*	*%*	*count*	*%*	*count*	*%*	*count*	*%*
move	move	17	20	45	6	21	3	25
at	at	20	15	34	5	18	2	17
away	away	39	13	30	4	14	5	42
at	away	29	5	11	10	35	0	0
away	at	13	11	25	9	32	5	41

The table shows that for mirroring episodes containing synchronized behavior (type 1), the behavior combinations indicating mirroring {move, move} and {at, at} have a high count, while the {away, away} count is low compared to the all category. The {move, move} combination is present 45% of the time in the first second of type 1 episodes, compared to 17% in the whole of the conversation. For mirroring episodes of type 2 the most striking difference in counts can be seen when comparing behavior combinations {away, away} and {at, away} between subcategories Lstart and Rstart. It seems that:

- {away, away} occurs normally 39% of the time, but when Lidewij is initiating a mirroring episode this is 14%. Rob's percentage in this context is close to normal (42%). So when both participants are gazing away, it is unlikely that Rob will follow Lidewij's behavior.

- {AT, AWAY} occurs 29% of the time normally, when Lidewij starts this is 35%, but when Rob initiates a type 2 mirroring episode this does not occur at all. So when Lidewij is gazing at Rob and Rob is gazing away it is unlikely that she will follow Rob's behavior.

This could mean that Lidewij and Rob have different 'triggers' for mirroring the behavior of the other.

4.7 Discussion

At the start of this chapter two questions about conversational behavior were stated: which behavior patterns occur and in which contexts. Following some examples found in literature, is was concluded that to model an Artificial Listener a rule base could be constructed based on behavioral patterns describing specific conversational processes. A recorded, task-oriented conversation from the gt2m data, was annotated in two stages:

1. transcriptions of eye-gaze, head movement and speech behavior,
2. annotations of dialogue acts, floor management and mirroring episodes.

First, some basic statistics were gathered from the annotation data. This analysis gave insight into the following characteristics of the conversation:

- how certain behaviors are distributed over time (timing),
- how long certain behaviors are (duration) and
- how much they are used (frequency).

The annotation data was then used in an analysis that incorporated using co-occurring behaviors of Speakers and Listeners. The use of co-occurrences enabled the study of interactive behaviors in specific contexts. Some patterns found were:

- specific to the one person vs the other,
- specific to the context of smooth floor transitions,
- nods and shakes in different contexts,
- specific to mirroring episodes.

If using more data and annotators, it is believed that the methods employed in this chapter will provide more reliable and significant data about interactive behaviors to be used in rule-based models of Artificial Listeners. Nonetheless, to at least give *some* concrete results, the following list contains a few patterns of head movements and gaze found in this chapter:

- During smooth floor transitions, a gaze at by one person is followed with a gaze at by the other *more often* within 0.5 seconds than after 0.5 seconds.
- During smooth floor transitions, gaze away behavior is followed by a gaze away *more often* after 0.5 seconds than within 0.25 seconds.

- Before a smooth floor transition, the person taking the floor gazes *more* at the other than after a smooth floor transition.

- Before a smooth floor transition, conversants gaze at each other *longer* than in the context of a back-channel.

- When Lidewij is listening, she and Rob gaze at each other much longer before a smooth floor transition than before a less smooth transition.

- When Rob is listening the period before a floor transition takes place, features almost no gazing away from each other when the transition is smooth, while with a less smooth transition gazing away is not uncommon.

- When Lidewij is looking away and Rob is moving his head, chances that Lidewij is going to use a back-channel signal is four times higher than when they both gaze away.

- If Lidewij gets the floor or is giving feedback Rob is nodding or shaking double time or more compared to other contexts.

Chapter 5

Generic modelling of listening behavior

In this chapter a proposal is presented for the modelling of the decision system for generating conversational behavior of an artificial listener. This design proposal deals with a general approach to model the internal processes of a sensitive artificial listener (SAL) agent, enabling the agent to exhibit human-like listening behavior. It will be explored how an Embodied Conversational Agent (ECA), such as SAL, can be equipped with a decision module to generate behaviors. Specific focus is on generating behaviors of gaze, backchannel continuers such as “aha”, “hm” and head movements. The implementation shows an extensible decision system that can easily be adapted by adding rule-like language constructs that specify the decisions to be made. Additionally, the design proposed here is also fit for outputting *action primitives* for the talking head on-line, and in real-time. Action primitives are representations of the behaviors to be generated, such as “nod” or “move side fast”, they can function as input for a 3D talking head such as RUTH.

Conversations between an ECA and a human subject are taken from the Sensitive Artificial Listener database [Roddy Cowie et al., 2005] to test the implementation. The idea is to use these videos and add a 3D talking head representing a SAL character as depicted in figure 5.1: This allows for offline comparison and analysis of different model specifications of a decision module that generates the action primitives for RUTH to perform.

Recently, some other proposals dealing with declarative modelling of conversational behavior have been presented e.g. in the works of [Stefan Kopp et al., 2006, Jina Lee and Stacy Marsella, 2006]. These articles both describe systems or frameworks that are capable of specifying the generation rules for conversational behavior, similar to the approach taken here. We will shortly review the similarities and differences. Similarities in [Jina Lee and Stacy Marsella, 2006]:

- Use of XML languages to specify the rules that are used to produce behavior.

- Behavioral and functional aspects serve as input, description of behavior to generate (action primitives) as output. Rules can be defined to link functional aspects of conversations with appropriate behaviors.

Figure 5.1: embodied SAL with human subject

Some differences are that these earlier works:

- Use standardized XML languages for the description of functional aspects (FML) and behavioral aspects (BML). The proposal presented here allows for any type of input or output.
- Put emphasis on concise external representations of functional and behavioral aspects, and clearly defined modules that host these representations. Here the focus is more on how these representations can be linked together, without specifying the modules themselves.

First the ideas behind the rule-like language constructs will be presented, and shortly reviewed in sections 5.1 and 5.2. Then the ideas are slightly adapted and made more concrete in section 5.3. In section 5.4 a prototype implementation is presented and it's output and further capabilities will be reviewed.

5.1 Model-based agent

Using the typology fromRussell and Norvig [2003, p.48-49] the basis for our agent architecture is a *model-based agent*. In this textbook current views on Artificial Intelligence are summarized. The model-based agent we refer to here, as presented in this textbook is a specific type of rational agent: an agent that "acts so as to achieve the best outcome or, when there is uncertainty, the best expected outcome". The rational agent concept is central to the approach taken in this book, and fits the general approach to specifying a decision module for SAL as presented in the remainder of this chapter. The model-based agent is a kind of agent, or autonomous computer program, interacting with it's environment using some kind of *knowledge base*. The environment is everything external to the agent. Interaction is achieved by using:

- sensor input from the environment and
- action output to the environment.

The assumption is that based on sensor input and an internal state, the agent decides which action to take. The following list sums up four basic processes of the agent that will be explained in more detail later:

1. the sensor input is *evaluated* and meaningful events are detected
2. meaningful events can be combined to *form* a new meaningful event
3. meaningful events may trigger a reaction from the agent, in other words some meaningful events will cause the agent to *act.* A meaningful event can be mapped to an action primitive.
4. the internal mental state of the agent influences the action selection process of the agent. Depending on the state certain action primitives may or may not be *selected* when a meaningful event occurs.

For the SAL project an operator, a human controlling some aspects of SAL, must be able to influence the agent's behavior as well, this makes the agent less autonomous, but allows for more flexibility in experimental settings. In a Wizard of Oz setup, for example, the operator can choose the appropriate verbal response.

So a total of five concrete classes (entities) were mentioned above: sensor input, operator input, meaningful event, mental state, action primitive. These entities are the generic building blocks that the author of a SAL can specify, and which the agent uses to produce behaviors.

5.1.1 input

Here are two kinds of input are assumed: sensor input and operator input. Sensor input is provided by the sensory body parts of the agent, e.g. digital video cameras for eyes. Other sources of input could be a microphone, or even a speech recognition module. Important to note is that the sensor input entity does specify the type of data, merely that it is input data gathered by the agent itself.

As mentioned the operator input is provided by the person operating SAL. This input will mostly be in the form of a command to directly be performed by the agent, such as DO_NOD or SAY('hello').

5.1.2 Events

Events are *internal signals,* as opposed to the external signals coming from the *input,* that cause the *state* of the agent to change (see 5.1.3). Events can be specified to be generated upon the reception of a combination of state, event and input.

The prefix meaningful is added to stress that (1) only events that contribute to reaching the goals of the agent should be taken into account, (2) these events carry this meaning or purpose by itself, i.e. by the name (label) for the meaning that makes sense to the author of the decision system. Examples of events for SAL are: *otherIsHappy*, *otherWantsFeedback* and *doLookPuzzled.*

5.1.3 State

The mental state is the memory of the agent, this can be any type of memory (short-term, long-term, et cetera). The prefix *mental* is added to stress that the state is internal to the agent's *mind.* This entity can be used by SAL to (1) keep track of some, earlier inferred, external state like the dialogue state, and (2) enable a black-board type of knowledge sharing among different modules and rules internal to the agent. Examples are: *waitForFeedback*, *selfEmotionIsHappy* et cetera.

5.1.4 Action

It is assumed that the intention of the agent is to reach it's goals by changing the environment with it's actions. Actions are eventually carried out by the *body* of the agent. The body may consist of output modules such as a speech synthesizer, a 3D talking head, electro-motor circuitry. The modules will be fed by action primitives generated by the rule-base. So generation of action primitives is done by the agent if a specific state is reached or a specific event is fired. Examples are: “nod”, “move side fast”, “blink”, et cetera.

5.1.5 Entity relations

In figure 5.2 an entity relationship diagram is presented which describes how the defined classes relate to each other. This is an initial proposal that will be reviewed later in this chapter.

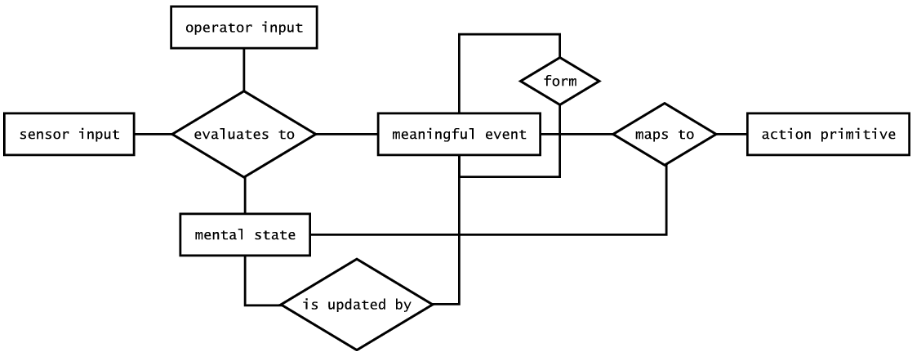


Figure 5.2: ERD of agent action selection mechanism
(read from left to right)

Note that this is a simplified model which ignores some relations. Some possibilities not included in the figure are OPERATOR INPUT adjusting the SENSOR INPUT, OPERATOR INPUT directly triggering actions, et cetera.

The question is now what constitutes the classes (which attributes are needed for the entities), and how do they relate and interact with each other in more detail? Before answering this question, let's take a look at some examples, and try to describe them using the entities and relations defined above.

5.2 Examples of model specifications

This is an exploratory exercise, after which some initial conclusions are drawn. Afterwards some adaptions are presented based on the findings here. Three different types of models found in literature will be presented and translated to fit the rule-like language constructs. In this way it can be checked if the framework (1) has clear semantics, (2) is flexible enough, and (3) if it can produce the desired output.

In this section three examples, based on rules or algorithms taken from literature, are shown. The examples mimic the original rules or algorithms, but they all use the entities and entity relations as described in the previous section. Three examples are shown:

- a probabilistic algorithm to control Listener's gaze
- a rule that produces a backchannel signal based on speech pitch values
- rules for turn-taking systems based on gaze, other behaviors, or state of a turn-taking model

5.2.1 Probabilistic model

In [Christopher Peters et al., 2005a] some measures of attention and interest based on eye gaze are used to decide if the speaker should continue the conversation. Focus in this work is on researching how behaviors of an artificial agent influences the level engagement ascribed to the agent by the (human) conversational partner. In one part of the system a probabilistic model is used to decide on the Listener's gaze behavior.

Looking at the model, two action primitives can be identified: look at and look away, also two states corresponding to these action primitives are present: L0 and L1. The states are necessary for the model to 'remember' what it is currently doing. Two constants are used that define the maximum gaze at or away time: Gl1 and Gl0. Variables can be stored in the mental state like the remain and transit probabilities: P(L0), P(L1L0), P(L0L1), P(L1). In the algorithm (Algorithm 2) presented in [Christopher Peters et al., 2005a] the probabilities are increased or decreased depending on the current gaze state (L0 or L1). This mechanism of updating probabilities of remaining in or transiting to a state, allows for controlled but not too predictable behavior. The next examples show how Algorithm 2 can be implemented in our model:

```
sensor_input(T) > mental_state(T) AND
mental_state(L0)
=>
meaningful_event(INCR_P_L0),
meaningful_event(INCR_P_L1L0),
meaningful_event(DECR_P_L1),
meaningful_event(DECR_P_L0L1)
```

Where sensor_input(T) represents the current time-stamp and mental_state(T) the previous. Also some update mechanism must be present which turns meaningful events into operations on the mental state:

```
meaningful_event(INCR_P_L0)
=>
mental_state(P_L0) add INCREMENT
```

Another part of the system (Algorithm 3) models the Speaker behavior. The level of interest of the Listener is used to decide whether it is desirable for the agent to continue the conversation. The level of interest of the listener (Li) is computed using a measurement of the attention of the Listener, which in turn is dependent on the gaze of the Speaker and the Listener. Here is an example of the EYECONTACT variable (Algorithm 1) and the effect on the perceived attention level, and how this could be implemented in our model:

```
sensor_input(EYE_DIR) approximates mental_state(MY_EYE_POS) AND mental_state(S1)
=>
meaningful_event(EYE_CONTACT)
```

and

```
(meaningful_event(EYE_CONTACT) integrate_over_time EFFECT_TIME) < EFFECT_LOW
=>
meaningful_event(DECR_EFFECT)
```

From these examples it can be derived that some additional relations must be defined like APPROXIMATES and INTEGRATE__OVER__TIME. Also the MENTAL STATE must be able to contain a memory of previous values of a certain variable, the same probably also goes for the other entities.

5.2.2 rule-based models

Maatman [R.M. Maatman, 2004] defines some rules for responsive behavior of a listening agent. These rules already resemble the rule-like structure proposed here, so it should be easy to convert them. The following examples define all of the rules needed to perform Maatman's first rule:

> *if the speech contains a relatively long period of low pitch then perform a head nod.*

This rule is based on Ward's algorithm [Ward and Tsukahara, 2000] that investigates how features from the speech signal influence a listener. The pitch feature calculated from the speech signal is a measure describing the tone of the voice. According to Ward low pitch can be determined by the 23th-percentile pitch. We can translate these requirements to rules that produce a head nod or other back-channel *action primitive*. First:

```
NOT mental_state(C_T_LOW_PITCH) AND
mental_state(OTHER_SPEAKING) check_over_time C_TIME_PITCH
=>
mental_state(C_T_LOW_PITCH) set TRUE,
mental_state(T_LOW_PITCH) set (sensor_input(PITCH) percentile (C_TIME_PITCH
0.23))
```

calibrates the low pitch threshold. Then the rules to produce the event:

```
sensor_input(PITCH) < mental_state(T_LOW_PITCH)
=>
meaningful_event(LOW_PITCH)


meaningful_event(LOW_PITCH)
=>
mental_state(LOW_PITCH) set TRUE
mental_state(LOW_PITCH) check_over_time 0.12 AND
mental_state(OTHER_SPEAKING) check_over_time 0.7 AND
(NOT action_primitive(BACK_CHANNEL)) check_over_time 0.8
=>
meaningful_event(DO_BACK_CHANNEL_0.7)
```

Another rule from Maatman:

> *if the human performs a head shake then mirror this head shake.*

This can be translated as follows:

```
sensor_input(HYAW_DIR) check_pattern '([-1]+[1]+) | ([1]+[-1]+)'
=>
meaningful_event(SHAKE)
```

The pattern string in the above rule is a regular expression that matches one or more -1s followed by on or more 1s or the other way around. This type of expression can be used on output of the HEADROT or HEADVAR programs presented in section 2.2, to detect a shake. Meaningful events such as SHAKE can then be used in an action selection procedure like this:

```
meaningful_event(SHAKE) AND mental_state(DO_MIRROR)
=>
meaningful_event(DO_SHAKE)
```

This last rule shows that depending on a specific *mental state* and a *meaningful event* the agent chooses to perform certain actions.

5.2.3 other rule-based models

In [Kristinn R. Thórisson, 2002] an implementation of some turn-taking rules is explained. The following lines represent the LOOK-PUZZLED-DURING-AWKWARD-PAUSE 'Overt Decision Module' in Thórisson's notation:

```
Look-puzzled-during-awkward-pause
EL: 1000 msec
BehaviorRequest: Look-puzzled
FIRE-CONDS: (AND (other-is-turned-to-me = T) (other-is-facing-me = T) (Time-since
Other-is-facing-me > 400))
RESTORE-CONDS: (Other-is-turned-to-me = F)
```

Here this could be represented as follows:

```
mental_state(OTHER_IS_TURNED_TO_ME) AND
mental_state(OTHER_IS_FACING_ME) AND
mental_state(OTHER_IS_FACING_ME) check_over_time 0.4
=>
meaningful_event(DO_LOOK_PUZZLED_1.0)
```

And the restore condition:

```
meaningful_event(DO_LOOK_PUZZLED_1.0) AND
NOT mental_state(OTHER_IS_TURNED_TO_ME)
=>
meaningful_event(RESET_LOOK_PUZZLED_1.0)
```

Starkey Duncan Jr. presents in [Starkey Duncan jr, 1975] a turn-taking system for speakers and auditors. To reach smooth exchange of turns from the auditor's point of view the following rules have to be satisfied in the notation presented here:

```
meaningful_event(PHONEMIC_CLAUSE_BOUNDARY) AND
NOT meaningful_event(WITHIN_TURN_SIGNAL) AND
NOT meaningful_event(GESTICULATION_SIGNAL) AND
meaningful_event(TURN_SIGNAL)
=>
meaningful_event(DO_SPEAKER_STATE_SIGNAL)

action_primitive(SPEAKER_STATE_SIGNAL) AND
meaningful_event(SPEAKER_SHIFT_TO_AUDITOR_STATE)
=>
mental_state(POSSES_SPEAKING_TURN) set TRUE
```

5.2.4 initial conclusions

It seems the rules defined using the entities and entity relation presented here, are able to describe different kind of behavior models, so this is a plus. Also, this makes it possible to use different models alongside of each other in the same system. Some things however remain unclear:

- The rule-like notation did not distinguish between user-defined functions and relations.
- The MEANINGFUL__EVENT entity seems to be used in different ways and for different purposes. It is used
  - as a mechanism to eventually produce an ACTION PRIMITIVE with the DO and RESET prefixes
  - as a mechanism to copy it's value to the MENTAL STATE, in this way it functions more like a sort of temporary state, and
  - to represent things that orgininated from the SENSOR INPUT in a more abstract manner.
- Nothing has been said about how ACTION PRIMITIVES are handled further. Sometimes it is assumed that actions are actually taken, other times the ACTION PRIMITIVE is used to check if an action was performed.
- How do the entities change over time, how are they updated?
- How can this be implemented in a computer program?

So, first some adaptions are proposed to tackle these problems.

5.3 Adaptions to the ERD

Looking back at the initial proposal and the exploratory exercises, we find that an important property of the entities is to expose what happens inside the agent.

This is mainly achieved by the use of MEANINGFUL EVENT and MENTAL STATE entities. However as the initial conclusions suggest the purpose of these entities and how they should be used is not clear yet.

The initial purpose of the MEANINGFUL EVENT entity was to represent some kind of *internal signal* for the agent itself to notify that something meaningful or important had happened. So things that belong in this entity are abstract descriptions of what can be found on the input, for example the detection of a SHAKE, but also more complicated assessments like SEEMS_HAPPY, which could be a combination of other events. One could suggest that the process of producing a MEANINGFUL EVENT in manner resembles the process of perception. On the other hand when defining rules for a FSA, MENTAL STATE in combination with SENSOR INPUT may produce a MEANINGFUL EVENT which in turn may update the MENTAL STATE. What this actually describes in terms of a FSA, is a state transition.

To summarize, other important issues are: the need for user-defined functions, an action selection procedure, and a short-term memory for MEANINGFUL EVENTs. The new ERD is split up for readability and attributes are added. The input entities have *time*, *label* and *value* attributes, MENTAL STATE has no *time* attribute because it always represents the state at present time. The MEANINGFUL EVENT doesn't have a value because it is a signal which is identified by it's label.

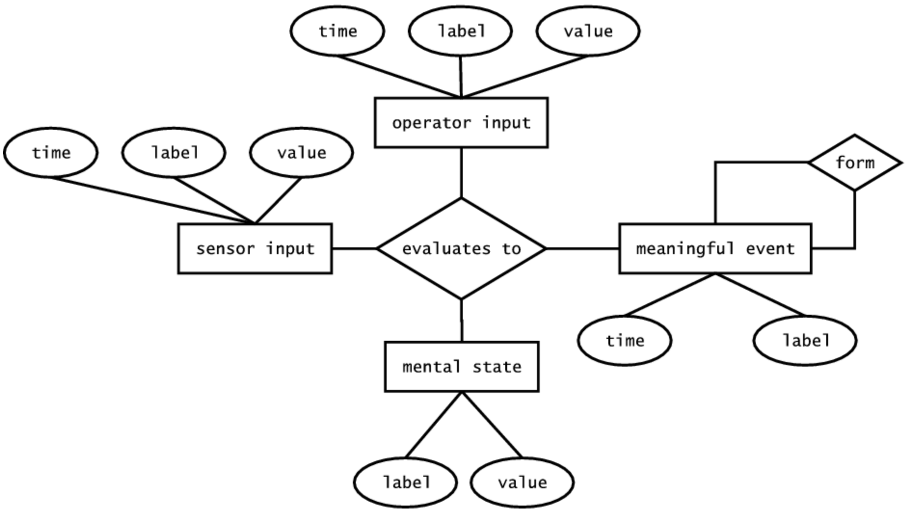


Figure 5.3: ERD of sensor input, operator input and meaningful event

If and when actions were supposed to be performed was also an issue in the first proposal. Now, based on MENTAL STATE and MEANINGFUL EVENT an action can be planned with the ACTION PLAN entity. The final step to actually perform the action is decided by rules having a LHS containing ACTION PLANs and MENTAL STATEs. This makes the action selection procedure more explicit in that it allows (1) planned actions to be cancelled and (2) mechanisms to choose between contesting actions.

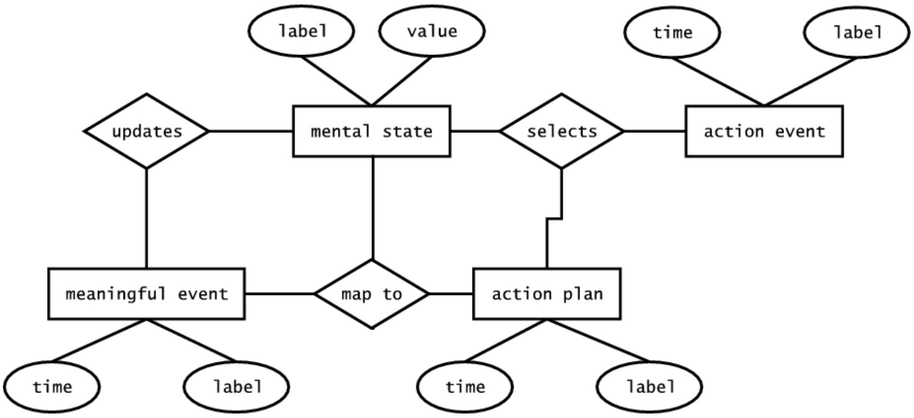


Figure 5.4: ERD of meaningful event, mental state, action complex and action event

The rule-like notation also needed some adaption. The ad-hoc notation used earlier allowed all kinds of constructs not necessarily linked to the original model. The next notation will make a distinction between user-defined *functions* now using the prefix notation, and *relations*. Relations like *evaluates to*, *updates, selects*, *forms* et cetera, are also depicted in the new ERDs. In the implementation these relations are used to produce new entities or update existing entities. Below is an example used earlier, now using the new notation.

```
check_pattern ( sensor_input(HYAW_DIR), ‘([-1]+[1]+) | ([1]+[-1]+)’ )
evaluate_to =>
meaningful_event(SHAKE)


meaningful_event(SHAKE) AND
mental_state(MIRROR)
map_to =>
action_plan(SHAKE)


action_plan(SHAKE)
select =>
action_event(SHAKE)
```

Another example:

```
check_value_over_time ( sensor_input(VOICED), 1, 0.7 )
evaluate_to =>
meaningful_event(OTHER_IS_SPEAKING_0.7)


meaningful_event(LOW_PITCH_0.12) AND
meaningful_event(OTHER_IS_SPEAKING_0.7) AND
check_value_over_time ( action_event(BACK_CHANNEL), FALSE, 0.8 )
map_to =>
action_plan(BACK_CHANNEL,0.7)
```

An update rule may look like this:

```
meaningful_event(INCR_P_LO)
update =>
mental_state(P_LO,add( mental_state(P_LO), INCREMENT ))
```

The user-defined functions, like CHECK__VALUE__OVER__TIME, are allowed to read the attributes of the entities, but not alter them. Furthermore there are two kinds of functions: *check functions*, which must return a Boolean value, and *calculate functions* which must return an integer or floating-point value.

5.3.1 Summary

In this section adaptions to the ERD and the rule-like notation for specifying behavior models was presented. To check whether this could work, an implementation is used to produce video output of a SAL character. The next section explores the capabilities of this implementation.

5.4 implementation: IAM

The Interactive Agent Modelling framework (IAM) is a Python implementation incorporating the ideas presented in the previous section. This section shows shortly how the implementation was realized, and what kind of results can be achieved.

5.4.1 Behavior rules, input and output

The behavior rules (model specifications) go into a file called 'model.xml', IAM also allows for defining so-called input and output processors that specify what is available at the SENSOR INPUT and OPERATOR INPUT entities in the model, and how ACTION EVENT entities are processed further. Output and input processors are defined in the 'processor.xml' file.

For now, the XML used in the 'model.xml' and 'processor.xml' files comply to the XML standard[1], but their sub-languages and the corresponding APIs are still in development and are thus not stable enough to be standardized. The rule-like notations found hereafter use a non-XML syntax for readability.

5.4.2 First example

Suppose we want to model the very simple behavior of mirroring nods of the speaker, we can then define some rules using the entities and relations described above, as we did before like this:

```
SensorInput(H_P_DIR) has_pattern '1111'
evaluate_to =>
MeaningfulEvent(NOD_DOWN)

MeaningfulEvent(NOD_DOWN)
plan =>
ActionPlan(MIRROR_NOD, 0.2s)

ActionPlan(MIRROR_NOD)
do =>
ActionEvent(DO_NOD)
```

[1] see Extensible Markup Language (XML) http://www.w3.org/XML/

First, a nod is detected on the sensor input with label H_P_DIR (head pitch direction). The user-defined function *has_pattern* checks for certain patterns over time and returns *true* if the given entity contains such a pattern over time. When the left hand side of the rule holds, a MEANINGFUL EVENT is produced. Upon detection of a MEANINGFUL EVENT with label NOD_DOWN an ACTION PLAN is produced using the *plan* relation. The first argument is a literal representing the action, the second is a time value indicating that this plan is made executed 0.2 seconds after now. Last, the action is performed using the *do* relation.

This example shows a rule which defines the behavior of, in the terminology of [Russell and Norvig, 2003], a *simple reflex agent*; no internal state is used, and the agent's reaction is based only on sensor input.

5.4.3 second example

The next example is more complex and will show some results that can be achieved using IAM with:

- the head movement tracker (see chapter 2.2), speech pitch and intensity detection[2] as input processors, and
- RUTH [Douglas DeCarlo et al., 2002] as output processor.

For testing purposes, data was gathered from a movie of the SAL project [Roddy Cowie et al., 2005] using the speech and head movement tools. Then, a model specification was applied to the data using IAM outputting a TMG file for use in RUTH. The TMG file serves as a low-level command specification for RUTH. RUTH was instructed to output images which were then glued together with the original movie. This process is depicted in figure 5.5.

The model.xml file used to produce the commands for RUTH and contains the following rule definitions:

- Mirror shakes and nods: whenever the other person does a shake or nod, mirror this behavior 0.2 seconds later.
- Random blinking: every second, 50% of the time produce a nod within 0.0 to 0.5 seconds.
- Fukayama gaze behavior: based on Atsushi Fukayama et al. [2002] amount of gaze 80%, mean duration 1.5, and some gaze points fitting RUTH.
- The pre-recorded speaking behavior: for lip movements and sounds. This behavior is fixed and taken from the transcriptions of the video.

In the generated movie (available online[3]) it is clearly shown that some of the 'behavior blocks', as specified in the model.xml file, collide sometimes with each other. For example when RUTH was looking away, shakes or nods are sometimes mirrored in between a gaze away period. So in practice a more sophisticated action selection procedure is required. Also additional checks may help. Other problems are:

[2] see `http://wwwhome.cs.utwente.nl/~rkooijma/#tools`

[3] see ruth-conv, model 1 `http://wwwhome.cs.utwente.nl/~rkooijma/ruth_conv/`

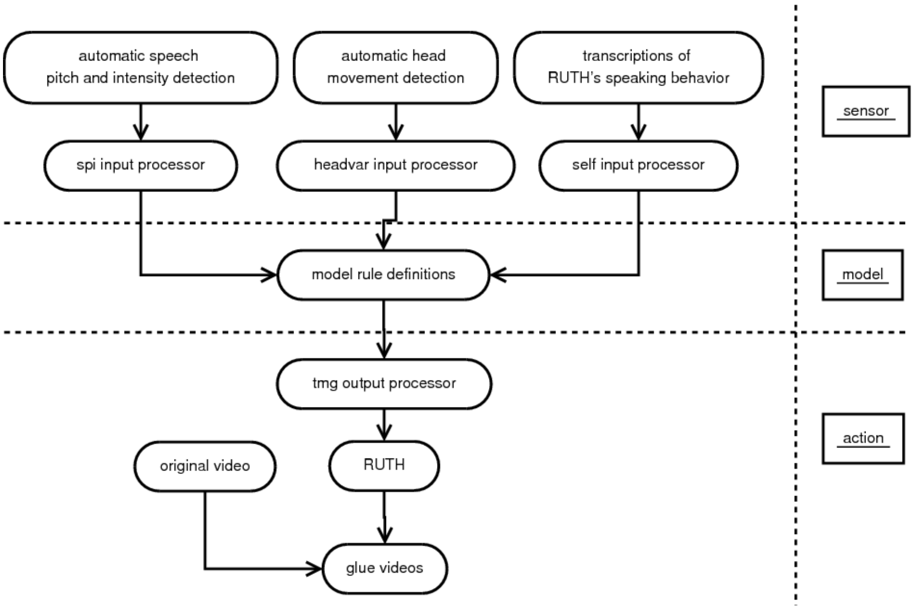


Figure 5.5: using IAM to create listening behavior

- with the recognition of shakes and nods by the head movement detector, and
- unnatural head movements due to the defined gaze behavior.

The default behavior for RUTH is that when the eyes are pointed in a new direction the head follows with a little delay, but this is not always necessary, and it results in too many head movements. One way to reduce this unwanted behavior is to use smaller angles, or to let the eyes move without the head following them. Also, looking up didn't seem appropriate all the time; it made it look like RUTH wasn't really paying attention. Note that we have not defined yet how we expect RUTH to behave. Also no exhaustive user-experience evaluation has been performed. To know how these models perform an evaluation on the impression the SAL character leaves on users watching the video could be done, but before evaluating the models, we need to find some models that perform acceptable enough to be put to the test.

Some improvements that could be implemented to make RUTH behave more *natural*:

- While speaking, speaker head movements can be synced with the speech accents, nothing has been done with this yet, but it should easy to implement.
- Blinking is random, but a little too much. Also some moments really 'ask' for blinking and gaze or not blinking and not gazing.
- Feedback sounds are not yet present (like "hm", "yeah", "aha" etc.)
- Different characters may have different gaze behavior.

- Reacting to speech accents (prosodic features) is possible, but not yet used.

Next, some of these improvements are implemented in a new model specification, then a separation of some of the behaviors is made to be able to improve them one by one.

5.4.4 Other models

Summarizing the characteristics of our first listener behavior model: mirror nods and shakes, stochastic gaze model, random blinks. Here, we will experiment with some other models. The rules for model 2 are based on the following requirements:

- If RUTH is speaking, don't do other behaviors.
- Use smaller gaze angles.
- Random blink less by altering the parameters to decide every 0.7 seconds to only perform a blink 20% of the time within 0.0 to 0.3 seconds.
- Don't perform a mirror nod or shake when gazing away.
- Fix a bug in the gaze model implementation.

These requirements for model 2 resulted in a new model.xml file for use in IAM. The result can be watched at the same URL as mentioned above by selecting model 2. Compared to the first model, model 2 has made some improvements but it seems some major issues are not fixed. The gazing away still looks artificial, and the mirroring of nods and shakes also doesn't look human-like. The blinking is still random, but the frequency is better now.

In the next step three models are devised that focus on one behavior aspect each: a better head movement mirroring model (3), a simple feedback model incorporating para-verbals (4), a fixed gaze model (5). The results of models 3, 4 and 5 are also available at the previously given URL. The following list summarizes the models:

- model 3: mirror substantial head movements of the speaker with a head movement that takes into account the magnitude of the speaker's head movement.
- model 4: Feedback model based on rule 1 and 2 from Maatman [R.M. Maatman, 2004]; produce feedback on low speech pitch or high speech intensity.
- model 5: Fixed Fukayama stochastic gaze model.

Interesting about model 3 is that because the magnitude of the perceived movement is taken into account the head movements look more natural than using the same nod and shake all the time, as in the other models. Model 4 has some problems with the timing of the feedback, this is mainly due to incorrect speech features detection. At some points it is also a matter of finding the 'right moment' to give a speech feedback, e.g. uttering 'aha' just after the speaker starts talking again doesn't seem appropriate, and could be perceived more as an interruption than feedback. Since with humans the head never is completely still, another improvement could be to add some small noise-like movements when no other head movement behavior is active.

5.5 Conclusions

The intention here was to present a uniform way to describe different kinds of conversational agent models, especially listening agents. After an short review some changes have been made to the initial proposal. The implementation of the proposal (IAM) shows that a listening agent can be equipped with:

- a fixed set of action primitives (repertoire),
- behavior rules that decide if and when these actions should be performed,
- input processors to enable perception, and
- output processors as actuators.

The videos show that IAM is capable of generating Listener behavior based on a variety of Listener models. This kind of output enables further experiments and evaluations of rule-based behavior models.

Chapter 6

Conclusions and recommendations

Let's first give a short summary of the different sections and their findings. Chapter 2 presented a head tracker and the notion of *elementary head movement* (EM). An EM described a head movement using the following features along three rotation axes: duration, direction and magnitude. EMs are thus giving a fine-grained account of the movements made, possibly enabling SAL to use this information to infer what a Speaker is communicating. So it allows the agent to *see* what happens, but also important, to react to what happens and to see how the Speaker reacts to what SAL itself does. Although an online survey of the head tracker with the HEADFEAT program, which measures the EMs, and a unigram language model, did not give that good results, it can be argued that using EMs to describe head movements is preferable to other annotation scheme's describing head movements.

In chapter 3 different aspects of face-to-face conversations are taken into account. The analyses in this chapter show that head movements made by conversants are used in a wide variety of contexts. In the analyses head movements are labeled with the following context categories: cognitive, social, linguistic, interactive and emotional. Furthermore, the chapter addresses how the functions of head movements can be linked to their forms. This allows for automation of a recognition and understanding module for SAL.

Chapter 4 focused on analyses of interactive behavior, and shows how aspects like timing play an important role in these kind of analyses. Furthermore it suggested that, if enough data is analyzed, it will be possible to construct rules that would help SAL in processes of floor management, mirroring, feedback and back-channeling. Some example rules are given extracted from the analyses performed in the chapter. The rules show that specific interactive contexts, such as smooth floor transitions have specific head movement and gaze patterns associated with it in the studied data.

If we look at these three chapters and try to view their contents and results as the components or ingredients for the construction of a SAL, the findings presented in chapter 5, allow for putting these ingredients together. The IAM framework presented in this chapter can connect input from e.g. the HEADFEAT program (chapter 2) to recognition rules, as presented in chapter 3, combined with rules of floor management from chapter 4, to a talking head such as RUTH to employ behavior fit for a Sensitive Artificial Listener. Also, in chapter 5, proof-of-concept is given by implementing a few rules for Listener head movements, gaze and para-verbal behavior, taken from literature, and outputting the result to a video.

Recommendations

In the introduction of this thesis it was stressed that the research presented here was of exploratory nature. The above sketched what we explored, but the question that remains is: what do we explore next, which direction should we go from here?

To be able to make SAL-like ECAs we need to know what these agents need to not only *hear*, but also *see*. In this thesis the notion of elementary head movement (em), was introduced to specify head movement forms. Key to this notion was that it captures features of the movement, in a rich but tangible way, so as to not neglect possibly meaningful variations and subtleties. So, which features are important for human perception of head movements? Which features of head movements contribute to the meaning of the movement in a specific context? Some leads are given in the work presented here, but further, possibly more domain-specific, experiments could be set up to find out the impact of features like speed, magnitude, duration on the human perception of head movements. Also, automatically perceiving head movements for the construction of a SAL is important in the quest for automatic recognition of the meaning of these behaviors.

What we did in this work is to try to identify the meaning head movements as used in face-to-face conversations. The analysis of chapter 3 showed that it may be possible to identify the functions of a head movement based on the head movement form. In the analysis it was assumed that aspects like cognition, social behavior, semantics, interaction and emotion, all play their role in determining head movement form. These aspects may be considered the context in which head movements need to be interpreted. What research dealing with the use of head movements in conversations could do, is on one hand try to isolate these contexts in experiments using the Wizard-of-Oz setup, as done in the SAL videos, but also set up experiments that focus on combining the different contexts, and try to find out how they complement each other.

In research such as that of [Kristinn R. Thórisson, 2002], it is shown that natural turn-taking using an ECA is feasible by separating communication processes from content. One of the issues highlighted in Thórisson research is that of timing. In chapter 4 behaviors in interactive contexts are studied with a fine level of detail with regard to the aspect of timing. The analysis in this chapter shows that behaviors of gaze and head movement, in contexts of e.g. smooth floor transitions compared to these behaviors in other interactive contexts, show different patterns. Here patterns were identified by co-occurrence counts of the behaviors of the two conversants. The analysis also differentiated between co-occurrences before and after a floor transition and showed that these parts of the conversation have different patterns associated with it. To analyze interactive processes in conversations timing aspects can be taken into account, which may yield new insights on how these processes are coordinated.

In the original SAL project, artificial characters were developed to see how people adapt emotionally to the characters. One could also try to do the exact opposite: let SAL adapt to the emotional state of it's conversational partner, and see what impression SAL leaves then. Adaption to the conversational partner is not studied in this thesis, but the work presented here does give some clues on which parameters of SAL could be adapted, and how. These parameters can be related to social aspects, but adaption may range from adaption to features of head movement behaviors such as timing, amplitude, duration, frequency et cetera, to adaption to the cognitive, or emotional state.

Bibliography

Adam Kendon. Some uses of the head shake. *Gesture*, 2(2):147–182, 2002.

Alessandro Duranti. The social ontology of intentions. *Discourse Studies*, 8(1): 31–40, 2006.

Allen Newell. *Unified theories of cognition.* Cambridge, Mass. : Harvard University Press, 1990.

Ashish Kapoor and Rosalind W. Picard. A Real-Time Head Nod and Shake Detector. In *Proceedings from the Workshop on Perceptive User Interfaces*, November 2001.

Atsushi Fukayama, Takehiko Ohno, Naoki Mukawa, Minako Sawaki, and Norihiro Hagita. Messages Embedded in Gaze of Interface Agents: Impression management with agent's gaze. *Conference on Human Factors in Computing Systems*, April 2002.

Berardina De Carolis Catherine Pelachaud, Valeria Carofiglio. Embodied Contextual Agent in Information Delivering Application. In *AAMAS '02*, July 2002.

Charles Goodwin. *Conversational Organization: interaction between speakers and hearers.* Acadamic Press, New York, 1981.

Christopher Peters, Catherine Pelachaud, Elisabetta Bevacqua, Maurizio Mancini, and Isabella Poggi. A model of attention and interest using gaze behavior. In *IVA*, pages 229–240, 2005a.

Christopher Peters, Catherine Pelachaud, Elisabetta Bevacqua, Maurizio Mancini, and Isabella Poggi. Engagement Capabilities for ECAs. *Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems, AAMAS '05 Workshop: Creating bonds with humanoids*, 2005b.

Dirk Heylen. A Closer Look at Gaze. *AAMAS Workshop on Creating Bonds 2005 (in press)*, 2005a.

Dirk Heylen. Challenges Ahead: Head movements and other social acts in conversations. *AISB 2005 - Social Presence Cues Symposium (in press)*, 2005b.

Doug DeCarlo and Matthew Stone. *The Rutgers University Talking Head: RUTH.* Department of Computer Science and Center for Cognitive Science, Rutgers, the State University of New Jersey, 2002.

Douglas DeCarlo, Corey Revilla, Matthew Stone, and Jennifer J. Venditti. Making Discourse Visible: Coding and Animating Conversational Facial Displays. *Computer Animation, 2002, Proceedings of*, pages 11–16, 2002.

Evelyn Z. McClave. Linguistic functions of head movements in the context of speech. *Journal of Pragmatics*, 32:855–878, 2000.

Hannes Vilhjálmsson and Stacy C. Marsella. Social Performance Framework. *Workshop on Modular Construction of Human-Like Intelligence*, 9 July 2005.

Hans Peter Graf, Eric Cosatto, Volker Strom, and Fu Jie Huang. Visual Prosody: Facial Movements Accompanying Speech. In *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition (FGR'02)*, 2002.

Harry Bunt. Context and Dialogue Control. *THINK Quarterly*, 3:19–31, 1994.

Harry Bunt. Dynamic Interpretation and Dialogue Theory. 1996.

D.K. Heylen. Head gestures, gaze and the principles of conversational structure. *International Journal of Humanoid Robotics*, 3:241–267, 4 November 2006.

James Allen and Mark Core. Draft of DAMSL: Dialogue Act Markup in Several Layers. http://www.cs.rochester.edu/research/speech/damsl/RevisedManual/, 22 September 1997.

Jina Lee and Stacy Marsella. Nonverbal behavior generator for embodied conversational agents. *IVA 2006*, pages 243–255, August 2006.

Jonathan Gratch, Anna Okhmatovskaia, Francois Lamothe, Stacy Marsella, Mathieu Morales, and R. J. van der Werf. Virtual Rapport. 2006.

Justine Cassell. *Embodied Conversational Agents*, chapter Nudge Nudge Wink Wink: Elements of Face-to-Face Conversation for Embodied Conversational Agents, pages 1–28. MIT Press, 2000.

Kristinn R. Thórisson. Natural Turn-Taking Needs No Manual: Computational Theory and Model, from Perception to Action. *Multimodality in Language and Speech Systems*, pages 173–207, 2002.

Kristinn R. Thórisson. A Mind Model for Multimodal Communicative Creatures & Humanoids. *International Journal Of Applied Artificial Intelligence*, 13: 449–486, 1999.

V. A. Jefferis Lakin, J. L. Chameleon Effect as Social Glue: Evidence for the Evolutionary Significance of Nonconsious Mimicry. *Journal of Nonverbal Behavior*, 27(3):145–162, 2003.

Language Archiving Technology. ELAN. http://www.lat-mpi.eu/tools/elan/, 12 April 2007.

Loredana Cerrato and Mustapha Skhiri. Analysis and measurement of head movements signalling feedback in face-to-face human dialogues. In Paggio P., Jokinen K., and Jönsson A., editors, *Proceedings of the First Nordic Symposium on Multimodal Communication*, pages 43–52, September 2003a.

Loredana Cerrato and Mustapha Skhiri. A method for the analysis and measurement of communicative head movements in human dialogues. *Proceedings of AVSP*, pages 251–256, 2003b.

M. Wooldridge. *An introduction to MultiAgent Systems*. John Wiley & sons, ltd, May 2001.

Marc Schröder and Roddy Cowie. Toward emotion-sensitive multimodal interfaces: the challenge of the European Network of Excellence HUMAINE. http://emotion-research.net, 2005.

M.H. Goodwin and C. Goodwin. Gesture and co-participation in the activity of searching a word. *Semiotica*, 1986.

Nicola Cathcart, Jean Carletta, and Ewan Klein. A Shallow Model of Backchannel Continuers in Spoken Dialogue. 2003.

Nicole Chovil. Discourse-Oriented Facial Displays in Conversation. *Research on Language and Social Interaction*, 25:163–194, 1991.

Catherine Pelachaud, Justine Cassell, Norman I. Badler, Mark Steedman, Scott Prevost, and Matthew Stone. Synthesizing cooperative conversation. In *Multimodal Human-Computer Communication, Systems, Techniques, and Experiments*, pages 68–88, London, UK, 1998. Springer-Verlag. ISBN 3-540-64380-X.

Steven M. Drucker R. Alex Colburn, Michael F. Cohen. The Role of Eye Gaze in Avatar Mediated Conversational Interfaces. Technical report, Microsoft Research, Microsoft Corporation, One Microsoft Way, 31 July 2000.

Rana Ayman el Kaliouby. Mind-reading machines: automated inference of complex mental states. Technical Report 636, University of Cambridge, May 2005.

R.M. Maatman. Responsive Behavior of a Listening Agent. Technical report, Institute for Creative Technologies, December 2004.

Roddy Cowie, Ellen Douglas-Cowie, and Cate Cox. Beyond emotion archetypes: Databases for emotion modelling using neural networks. *Neural Networks*, 18: 371–388, March 2005.

Rosalind W. Picard. *Affective Computing*. Affective Computing, 1997.

Stuart J. Russell and Peter Norvig. *Artificial Interlligence, a modern approach*. Prentice Hall, 2 edition, 2003.

Kim Silverman, Mary Beckman, John Pitrelli, Mori Ostendorf, Colin Wightman, Patti Price, Janet Pierrehumbert, and Julia Hirschberg. TOBI: a standard for labeling English prosody. *ICSLP*, pages 867–870, 1992.

Starkey Duncan jr. On the structure of speaker-auditor interaction during speaking turns. In *Language in Society*, volume 2, pages 161–180, 1975.

Stefan Kopp, Brigitte Krenn, Stacy Marsella, Andrew N. Marshall, Catherine Pelachaud, Hannes Pirker, Kristinn R. Thórisson, and Hannes Vilhjálmsson. Towards a Common Framework for Multimodal Generation: The Behavior Markup Language. *6th International Conference on Intelligent Virtual Agents*, August 2006.

Thanarat Horprasert, Yaser Yacoob, and Larry S. Davis. Computing 3-D head orientation from a monocular image sequence. *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pages 242–247, October 1996.

N. Ward and W. Tsukahara. Prosodic features which cue back-channel responses in English and Japanese. *Journal of Pragmatics*, 23:1177–1207, 2000.

Yukiko I. Nakano, Gabe Reinstein, Tom Stocky, and Justine Cassell. Towards a Model of Face-to-Face Grounding. In *41st Annual Meeting of the Association for Computational Linguistics*, July 2003.

Yuri Iwano, Shioya Kageyama, Emi Morikawa, Shu Nakazato, and Katsuhiko Shirai. Analysis of Head Movements and Its Role in Spoken Dialogue. In *Proceedings of ICSLP*, 1996.

Z. Ruttkay, C. Dormann, and H. Noot. Evaluating ECAs: What and how? In *Proceedings of the AAMAS02 workshop on "embodied conversational agents: let's specify and evaluate them!"*, 2002.

Appendix A

Gesprek Tussen 2 Mensen video data (gt2m)

This part of the thesis reports about the video collection called gt2m, which is a Dutch acronym for “conversation between two people”. The video data now consists of 8 conversations made by 4 dyads each having:

- one small-talk session, in which they could talk about anything
- one task-oriented session, in which they were given an assignment.

In the task-oriented conversation the participants were asked to work together on the task of formulating three questions for prime minister Balkenende. The participants have to cooperate to come up with the questions to be written down on paper.

A.1 recording setup

The participants sit in a small angle from each other presented as a top-view in figure A.1. Each of the participants had a camera directed to capture mainly the head, the upper body, arms and hands. Figure A.2 shows a snapshot of the two camera recordings mixed to one video image. After the recordings both camera views (of the left and right person) were put together to make one movie of the conversation. A snapshot of such a movie can be seen in figure A.2.

The average time of the conversations is about 12 minutes. It has to be noted that sound and lighting quality are not too good. For the sound also .wav files were constructed with a little better quality.

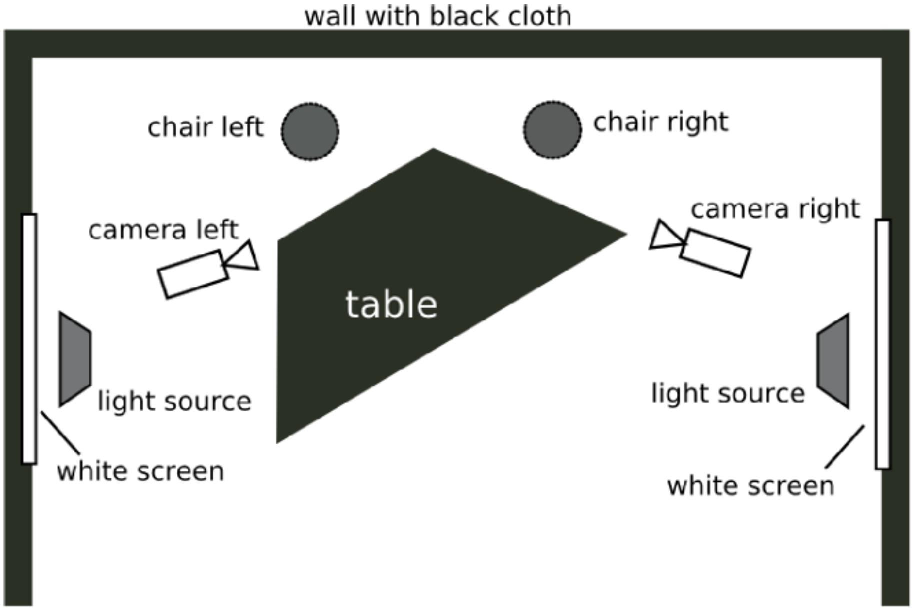


Figure A.1: recording room setup

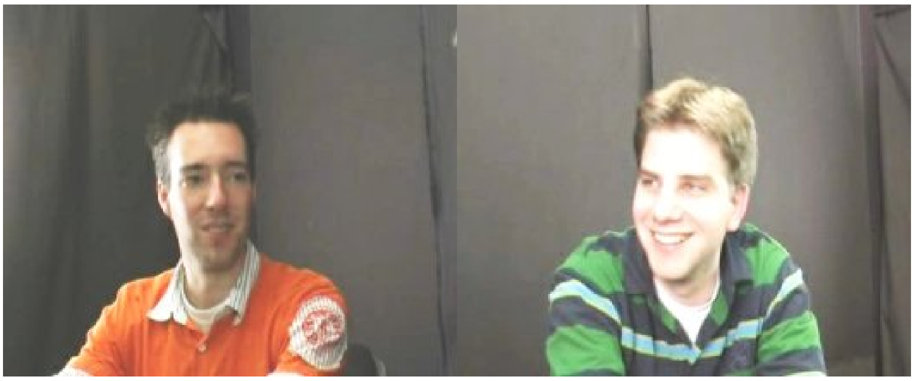


Figure A.2: snapshot from the GT2M data

Appendix B

A simple head movement tracker using OpenCV

Here, report is made about a simple head movement tracker developed for the SAL project, and for the analyses of head movements in conversations. The tracker is not intended to robustly and accurately capture rigid head rotations, but provides a simple and very light-weight solution without the need of more complex algorithms, and is, to some extend, usable for real-time recognition of head movements to be made during human-computer conversations.

B.1 Automatic head movement detection

The tracker consists of four programs that can be linked together with a pipe, connecting the output of one to the next. The four programs are:

- HMD: plays a movie file and tracks user selected points
- HEADROT: calculates indications of rotation angles given the position of the tracked points
- HEADVAR: produces filtered rotation angle velocities given indications of rotation angles
- HEADFEAT: segments the data into movement segments, and extracts some features of the movement.

B.2 hmd program

Because a practical, free head tracker capable of using monocular video images as input was not directly available, a simple program using the OpenCV library[1] was devised. From [Rana Ayman el Kaliouby, 2005] and [Yuri Iwano et al., 1996] it was inferred that tracking the position of the eyes and nose could be used to calculate pitch, yaw and roll (see B.3 for explanation of these terms). Assuming the eyes and nose in the footage were distinct enough to be tracked with the

[1] see OpenCv library `http://opencvlibrary.sourceforge.net/`

OpenCV function cvCalcOpticalFlowPyrLK, a C++ program (hmd) was made using this function, and outputting the data to a CSV file or pipe. The program works as follows:

- Read and show frames of the footage. When the user finds the participant in a fairly neutral position, let the user pause the movie. And go to step 2.

- Let the user select left eye, right eye and nose as well as two points not on the head (e.g. on the shoulders).

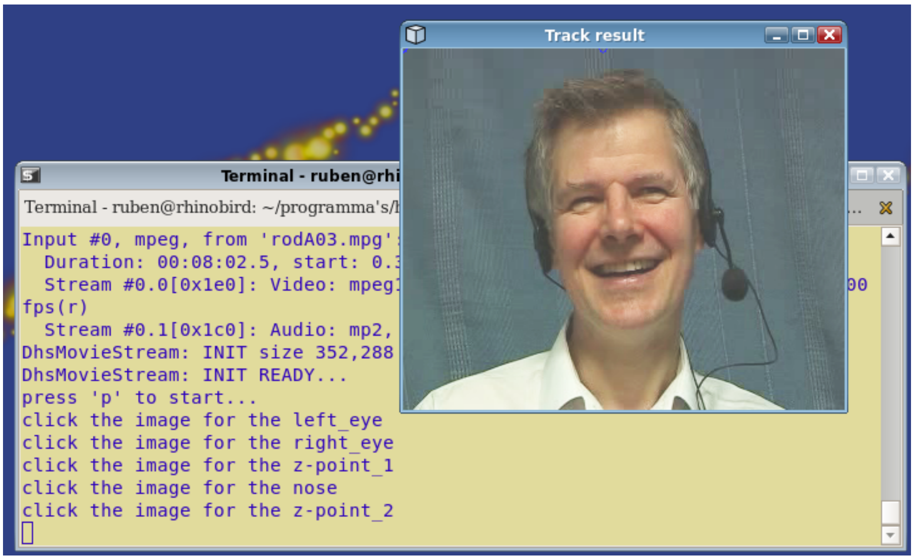
- Each selection is searched for ten points that can easily be tracked. The search is performed by the OpenCV function cvGoodFeaturesToTrack(...).

- Unpause the movie and perform the cvCalcOpticalFlowPyrLK(...) function to each set of the points in the following frames.

- Append the x- and y-coordinates of the five points to a CSV file, each frame until stop.

The function cvGoodFeaturesToTrack, in short, looks for points in a given region of the image that are surrounded by distinct pixel 'corners' or color combinations. The cvCalcOpticalFlowPyrLK function compares two video frames and gives the most likely coordinates of a point in the second frame given a point in the first frame.

Now the left eye, right eye and nose as well as two 'zero points' are tracked with ten points each, an average position can be taken as the center, to add a little robustness. The figures B.1 and B.2 show screenshots of the program in action. The yellow dots represent the average positions.



```
Input #0, mpeg, from 'rodA03.mpg':
  Duration: 00:08:02.5, start: 0.3
  Stream #0.0[0x1e0]: Video: mpeg1                    00
fps(r)
  Stream #0.1[0x1c0]: Audio: mp2,
DhsMovieStream: INIT size 352,288
DhsMovieStream: INIT READY...
press 'p' to start...
click the image for the left_eye
click the image for the right_eye
click the image for the z-point_1
click the image for the nose
click the image for the z-point_2
```

Figure B.1: screen-shot of selection of the track points

The output of the hmd program consists of 5*10=50 x,y coordinates, that can be used by the headrot program to calculate the rotation angles.

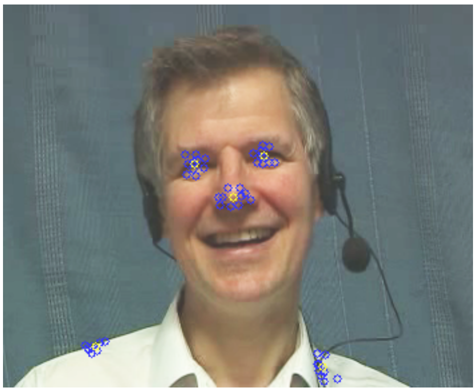


Figure B.2: screen-shot of feature point tracking

B.3 model

Figure B.3 shows how the tracked points relate to the rotation of the head. The two 'zero points' act as a reference and are not shown in the figure. In the model left, right eye and nose are connected to a rotation point in the upper neck. This rotation point is considered the origin of a three dimensional Euclidean space. The question now is: how can the coordinates of the points result in rotation angles around the x-axis (pitch), y-axis (yaw) and z-axis (roll)?

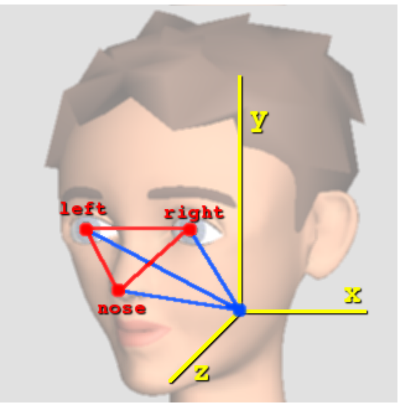


Figure B.3: feature points in head model

B.3.1 headrot: x,y-coordinates to rotation angles

To exactly calculate the rotation angles, a 3-D head model can be mapped onto the tracked feature points, also a stereo camera could be used to add depth information to the video frames. However, because the footage is not recorded using a stereo camera, and it is assumed that high accuracy of the angle calculations is not necessary to be able to interpret them, three formula's are devised that give an *indication* of the rotation angles. Figures B.4, B.5, B.6 show the three rotation angles φ to be calculated.

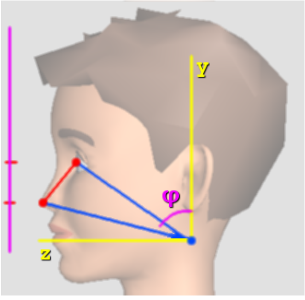


Figure B.4: pitch

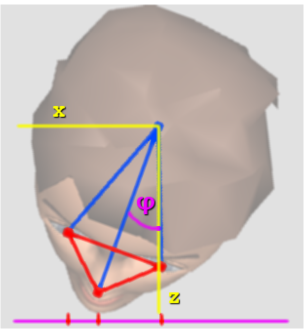


Figure B.5: yaw

The general rationale used to construct the formulas is that the head of the participant has a constant size and is projected onto the two dimensional plane of the video image. The red and blue lines in the figures have the same length during the tracking. Also, throughout the calculation the nose length is kept at a constant length. The observed distances between the points in the video image however may change because:

1. the head moves towards or from the camera (translations), and
2. the head rotates.

The magenta colored line in the pictures shows where the locations of the points is projected to on the video image. Now for each axis table B.1 shows for which value changes the rotations are most sensitive in the indicators column.

axis	*indicators*	*dependency*
x	nose length	y-rotation, z-translation
y	distance between right and left eye	x- and z-rotation, z-translation
z	corner of line through eyes and a horizontal line	x- and y-rotation

Table B.1: information about the angles

The indicators are inferred from the model B.3 and the figures B.4, B.5, B.6. All rotation angles may have a dependency on each other, this means that

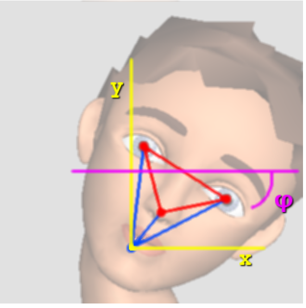


Figure B.6: roll

the x-rotation will play a subtle role in the value of the y-rotation when using the indicators. The z-translation, movement to or away from the camera, influences the measured, projected distances of all indicators. Some other important shortcomings are the dependence on: the exact placement of the track points, the distance from the camera, and the interpersonal differences in face geometry. These problems are partly solved by using calibration and normalization. Calibration is performed on a neutral head position for ten subsequent frames. Then the average nose bridge length, the distance between the zero points and the distance between the left and right eye are stored to compensate for the initial deviations. Despite the shortcomings, the indicators are used to construct the following formulas:

Table B.2: formulas to calculate rotation angles.

$$\varphi_x = \arcsin(1 - \frac{nl}{nl_0})$$
$$\varphi_y = \arcsin(1 - \frac{ed}{ed_0})$$
$$\varphi_z = \arctan(\frac{ed_y}{ed_x}) - \varphi_{0_z}$$

where

$$nl = d(nose, meye) \cdot zf$$
$$ed = d(reye, leye) \cdot zf$$
$$zf = \frac{zf_0}{d(zp1, zp2)}$$

The variables sub-scripted with 0 are determined at calibration time. The d(..,..) function represents the Euclidean distance function. Left-over are the variables:

- *nose*: coordinates of the tracked feature point at the nose tip,
- *reye*, *leye*: right and left eye coordinates,
- *meye*: coordinates in the middle between the eyes,
- *zp1*, *zp2*: the zero points.

The formulas can be used if one assumes:

1. a neutral position (straight to the camera) at calibration time

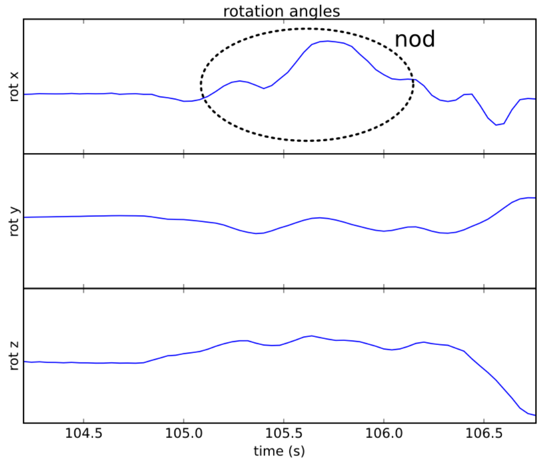
2. mainly small angles are to be measured

3. the nose does not protrude from the head; i.e. the z-coordinate of the nose tip is equal to the z-coordinates of the eyes in a neutral head pose.

Another solution resembling this one, but probably is more accurate in recovering the rotation angles of the head from monocular video images is described in [Thanarat Horprasert et al., 1996]. Like the solution presented here [Rana Ayman el Kaliouby, 2005] also does not exactly measure the angles, but uses ratio's of distances between tracked feature points. A completely different approach is to use active appearance models (AAMs) or some other model based fitting algorithm.

Since the output of HEADROT is further processed by HEADVAR and HEADFEAT to extract features of movement segments, exact rotation angles are not necessary. To give an impression however of some tracked movements the following figures B.7, B.8 and B.9 show some typical movements observed in available video data.

Figure B.7: nod measured with HEADROT

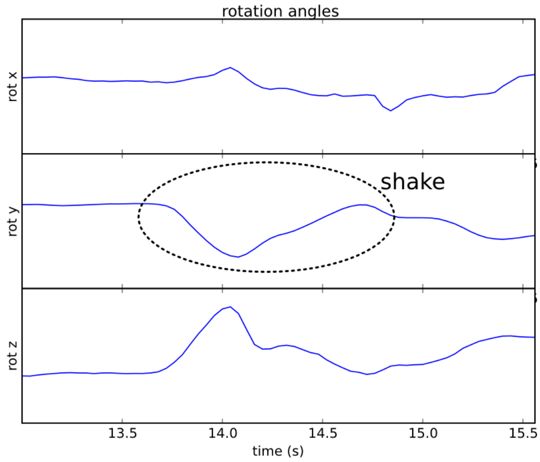


The figures show for each rotation angle the indication calculated by the formulas. A nod (figure B.7) results in rotation angles around the x-axis. This can be seen in the figure as an amplitude change in the rot-x signal. First the angle gets bigger, meaning a upward movement of the head, then it gets smaller again. Figure B.8 shows the angle indications of a lateral movement of the head. This mainly influences the rot-y amplitude, but as can be seen in the figure, also rot-z is influenced. Using the formulas, particularly for this fragment the head shake also causes rot-z changes over time because at the start of the fragment the head already had a non-zero rotation angle around the z-axis and, as noted earlier, this is a short coming of the formulas.

B.3.2 headvar: angles to rotation speed features

A movement can be defined to be a *change in position*, since our interest lies in measuring head *movements* the HEADVAR program calculates the change in

Figure B.8: shake measured with HEADROT



angles between two consecutive frames, or in other words the angle velocities.

First the output of HEADROT is smoothed to reduce noise using the moving average over a window of three frames. Then the following features are calculated for each axis:

- angle velocity,
- direction.

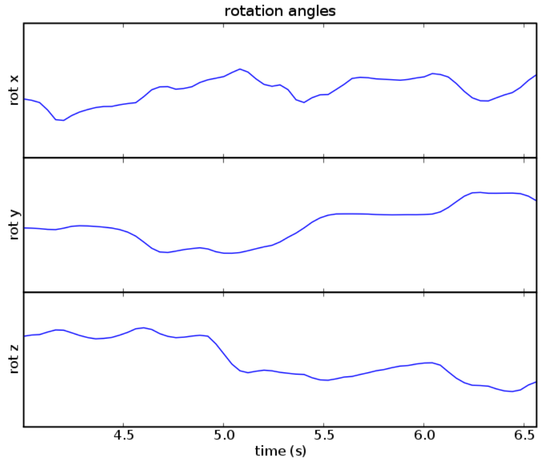
The direction feature has the values -1 or 1 depending on the direction of the movement. A value of 0 is returned if the angle velocity is considered too small. The thresholds can be configured manually for each axis. In the HEADFEAT program the direction value is used to determine segments in the movement data. Figures B.10, B.11 and B.12 show the same fragments as before, but now the output of HEADVAR is plotted.

B.4 headfeat: rotation features to segment features

As stated before, the human head will move constantly during a conversation. To analyze these movements the data must be segmented into some kind of *unit of analysis*. In this project the smallest unit of analysis is defined: an elementary head movement (EM). An EM is a determined by a movement (sequence of high angle velocities) along one axis in one direction. It has a:

- *duration* in seconds,
- cumulative *direction*: sum of HEADVAR directions in this segment,
- *magnitude* category: a value ranging from tiny to small to medium to big, and

Figure B.9: complex movement measured with HEADROT



- a *shape* vector describing the progression of the movement along this segment.

The HEADFEAT program finds EMs in the output of HEADVAR by looking for start and end patterns in the HEADVAR direction values over time. When it has found a segment the features mentioned above are calculated. More details on the segmentation process and the feature calculation are found in section 2.3.

Tables B.3, B.4 and B.5 show the output of HEADFEAT on the same fragments used before.

Table B.3: segment containing nod from HEADFEAT

time	*x*						*y*						*z*					
	dur	*dir*	*mag*	*shape*			*dur*	*dir*	*mag*	*shape*			*dur*	*dir*	*mag*	*shape*		
105.48	0	0	0	0	0	0	0.32	8	4	0.49	0.75	0.69	0	0	0	0	0	0
105.52	**0.08**	**-2**	**1**	**1**	**1**	**0.6**	0	0	0	0	0	0	0	0	0	0	0	0
105.56	0	0	0	0	0	0	0	0	0	0	0	0	0.12	-3	1	0.74	1	0.74
105.76	0	0	0	0	0	0	0	0	0	0	0	0	0.16	4	2	0.77	0.85	0.56
105.8	0	0	0	0	0	0	0.28	-7	4	0.6	0.83	0.57	0	0	0	0	0	0
105.84	**0.28**	**7**	**4**	**0.65**	**0.86**	**0.57**	0	0	0	0	0	0	0.08	-2	1	1	1	0.47
106.12	0	0	0	0	0	0	0.28	7	4	0.59	0.81	0.59	0	0	0	0	0	0
106.16	**0.32**	**-8**	**4**	**0.63**	**0.85**	**0.57**	0	0	0	0	0	0	0.16	-4	2	0.77	0.81	0.53

Table B.4: segment containing shake from HEADFEAT

	x						*y*						*z*					
time	*dur*	*dir*	*mag*	*shape*			*dur*	*dir*	*mag*	*shape*			*dur*	*dir*	*mag*	*shape*		
14.16	0.28	7	2	0.41	0.66	0.68	**0.4**	**10**	**4**	**0.62**	**0.81**	**0.52**	0	0	0	0	0	0
14.32	0.16	-4	3	0.86	0.71	0.39	0	0	0	0	0	0	0.2	-5	4	0.7	0.89	0.67
14.44	0	0	0	0	0	0	0	0	0	0	0	0	0.08	2	1	1	1	0.55
14.56	0.12	-3	1	0.74	1	0.65	0	0	0	0	0	0	0	0	0	0	0	0
14.72	0.04	-1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
14.8	0	0	0	0	0	0	**0.64**	**-16**	**4**	**0.6**	**0.59**	**0.47**	0	0	0	0	0	0
14.84	0	0	0	0	0	0	0	0	0	0	0	0	0.36	-9	4	0.59	0.67	0.59
14.96	0.08	-2	2	1	1	0.57	0	0	0	0	0	0	0.12	3	2	0.71	1	0.73
15	0	0	0	0	0	0	0.2	5	4	0.68	0.81	0.64	0	0	0	0	0	0

Table B.5: complex movement from HEADFEAT

	x						*y*						*z*					
time	*dur*	*dir*	*mag*	*shape*			*dur*	*dir*	*mag*	*shape*			*dur*	*dir*	*mag*	*shape*		
4.84	0	0	0	0	0	0	**0.72**	**8**	**4**	**0.2**	**0.38**	**0.82**	0.2	-2	2	0.51	0.61	0.76
5	0	0	0	0	0	0	0.12	-3	2	0.75	1	0.6	0	0	0	0	0	0
5.16	0	0	0	0	0	0	0.12	3	2	0.84	1	0.6	0	0	0	0	0	0
5.2	**0.84**	**16**	**4**	**0.58**	**0.51**	**0.52**	0	0	0	0	0	0	0	0	0	0	0	0
5.4	0	0	0	0	0	0	0	0	0	0	0	0	0.48	-5	4	0.9	0.81	0.2
5.48	0.28	-6	4	0.49	0.63	0.52	0	0	0	0	0	0	0	0	0	0	0	0
5.64	0	0	0	0	0	0	0	0	0	0	0	0	0.12	-3	1	0.78	1	0.69
5.68	0	0	0	0	0	0	**0.52**	**-13**	**4**	**0.45**	**0.71**	**0.66**	0	0	0	0	0	0
5.76	0.28	7	3	0.39	0.8	0.72	0	0	0	0	0	0	0	0	0	0	0	0
6.12	0	0	0	0	0	0	0	0	0	0	0	0	**0.44**	**11**	**3**	**0.59**	**0.68**	**0.51**

Figure B.10: nod measured with HEADVAR

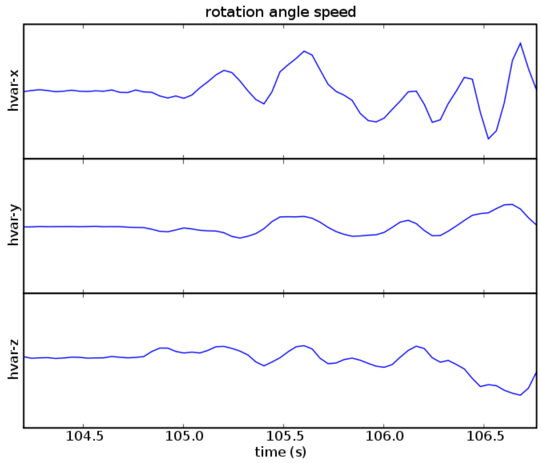


Figure B.11: shake measured with HEADVAR

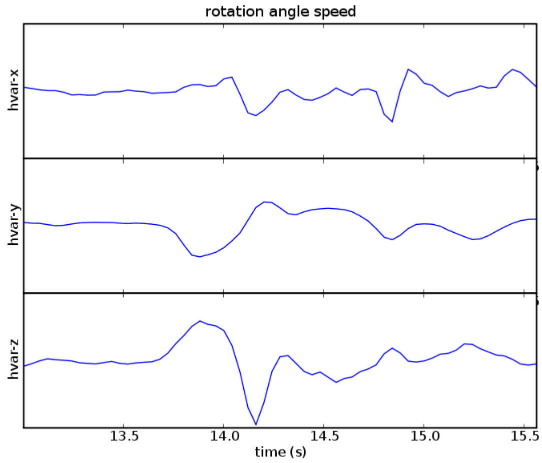
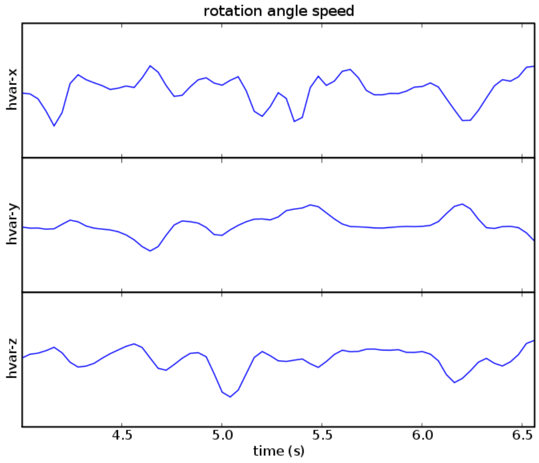


Figure B.12: complex movement from HEADVAR


rotation angle speed
hvar-x
hvar-y
hvar-z
time (s)