AAL-2009-2-049, ALIAS

D3.7

Dialogue System Updated to User's Needs

Due Date of Deliverable	2012-05-31
Actual Submission Date	2012-07-26
Workpackage:	3.7
Estimated Staff Months:	?
Dissemination Level:	Public
Nature:	Report
Approval Status:	Final
Version:	v2.0
Total Number of Pages:	29
Filename:	D3.7-MMK-DM-v2.0.pdf
Keyword list:	Dialogue System, Dialogue Manager, Human-Machine Communication

Abstract

One of the main components of the ALIAS robotic platform is the Dialogue System. Each input and output that is triggered by a user is supervised by the Dialogue System, which communicates with all other modules integrated on the robot. Person detection and identification are closely coupled to the Dialogue System, as they are essential for dialogue situations. In addition, ALIAS uses emotional facial expressions to give feedback to the users. In this deliverable, the technical details of the Dialogue System are described in detail. In addition, some other modules that are closely coupled to the dialogue manager are presented: person detection and speaker identification systems as well as a module for displaying emotions. Two use-case scenarios are described to show how the dialogue system is applied.

History

Version	Date	Reason	RevisedBy
1.0	2012-07-16	created [MMK]	Juergen Geiger
1.1	2012-07-17	reviewed [IUT]	Jens Kessler
1.2	2012-07-18	updated [MMK]	Juergen Geiger
1.3	2012-07-25	reviewed [FHG]	Niko Moritz
2.0	2012-07-26	updated [MMK]	Juergen Geiger

Authors

Partner	Name	Phone / Fax / Email
MMK	Juergen Geiger	Tel: +49 89 289 28543 Fax: Email: geiger@tum.de
COG	Michael Pasdziernik	Tel: +49 241 4010208 12 Fax: Email: mpasdziernik@cognesys.de
EUR	Ravichander Vipperla	Tel: Fax: Email: Ravichander.Vipperla@eurecom.fr
IUT	Jens Kessler	Tel: +49 3677 69 4170 Fax: Email: jens.kessler@tu-ilmenau.de

Table of Contents

1 Introduction

The primary objective of the ALIAS project is to develop a mobile robot platform that is designed to assist elderly users and people in need of care to continue independent living with minimal support from carers. The functionalities of the robot platform will include among other services, the ability to interact with users, monitor their well being and, if necessary, provide cognitive assistance to them in everyday situations.

One important system compoment of the ALIAS robotic platform is the dialogue system and its dialogue manager (DM). The dialogue system is the central hub of ALIAS and connects all other components of the platform. On the one hand, any communication between system components is handled by the dialogue system. Therefore the DM can control all other modules. On the other hand, any user input and output is handled by the DM, whereby the DM acts as a natural language understanding system. For each possible input from the user, the DM is responsible for the interpretation and the decision about the next action of the system.

Other system components that are discussed are a module for person detection, a speaker identification system and a system for displaying emotions with the robotic platform. In order to begin a dialogue with a user, first the robot has to detect the person and determine his or her position. Therefore, two different approaches are used. Face detection is used to find human faces with a camera mounted on top of the robotic head. In addition, a laser range finder is used to detect leg pairs. A combination of these two person detection modules leads to a more robust and reliable person detection. To identify the dialogue partner, speaker identification is employed. During a human-machine interaction, this is a natural non-intrusive way of determining the user's identity. Finally, in order to give appropriate natural feedback to the dialogue partner, ALIAS uses his robotic head to display emotions. This enhances the way of the communication due to a higher degree of familiarity with the robot.

This deliverable describes the functionality of the dialogue manager and its adaptation to the user's needs. For example, a module for displaying emotions was developed to satisfy the wishes for natural communication with the robot. This module, in addition with some other related modules, will be described in this document. In Chapter 2, the architecture of the DM and interfaces to other modules are described. The person detection module that is based on camera-based face detection and laser-based leg detection is described in Chapter 3. Chapter 4 deals with the implementation of the ALIAS speaker identification module. Chapter 5 describes how the robotic system ALIAS can display emotions to its dialogue partner. Finally in Chapter 6, it is described how the dialogue manager is applied in two exemplary use-case scenarios. A summary and conclusions are given in Chapter 7.

2 Dialogue Manager

This chapter describes the architecture of the Dialogue Manager (section 2.1) and interfaces to the other robot modules (section 2.2).

2.1 Architecture

The Dialogue Manager is the central communication and decision unit of the robot. It has two main parts: the Dialogue Manager Communicator (DM Communicator) and the Dialogue Manager Core (DM Core). Figure 2.1 shows the architecture.

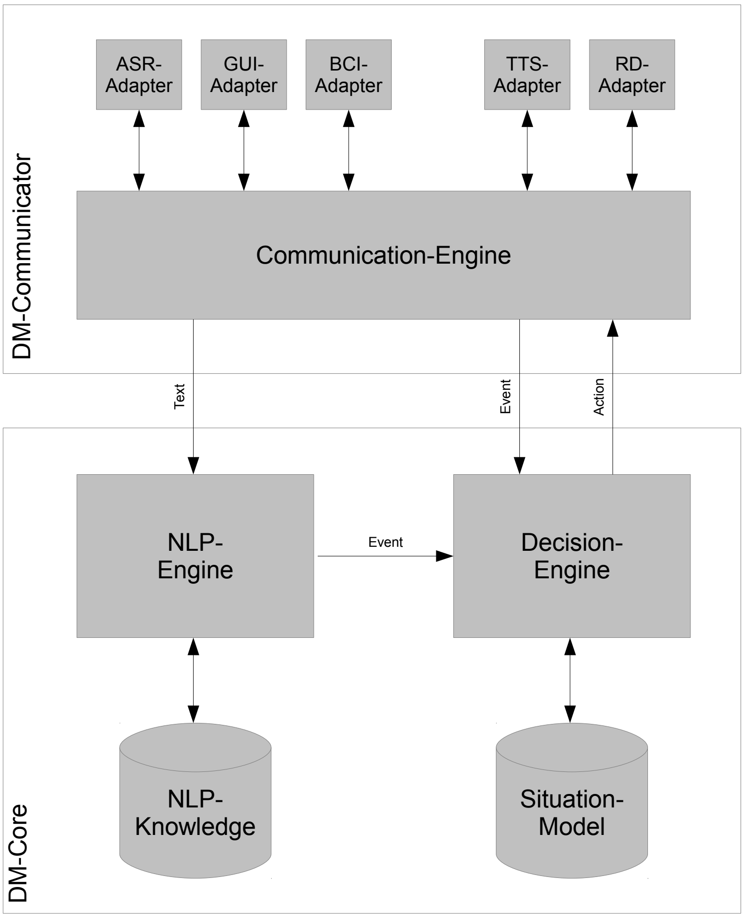


Figure 2.1: Dialogue Manager architecture

The DM Communicator manages the communication between the DM Core and the other components of the robot. Due to the integration of different modules with different architectures on one platform, various protocols and data formats need to be handled by the DM (see section 2.2). The Communication-Engine of the DM Communicator uses different adapters to abstract from this diversity by translating different input signals into a uniform event format and by translating DM Core action decisions into different outgoing signal formats.

The DM Core encompasses two major components, the natural language processing engine (NLP-Engine) and the Decision Engine. The NLP-Engine translates spoken user input provided as text by the ASR component into events that are then further processed by the decision engine. The Decision-Engine receives events from the NLP-Engine and from the DM Communicator and decides on the behaviour of the robot by choosing the next action. To determine the appropriate action, the Decision-Engine takes into account the current situation, which is based on the history of past events.

The DM Core uses two knowledge databases. The NLP knowledge base provides general information regarding the world and the life of elderly people and the translation of natural language to abstract event entities. The situation model knowledge base stores information needed by the Decision-Engine to assess the next situation and to determine the next action with respect to the current situation.

Both the DM Communicator and the DM Core are implemented in Common Lisp. The DM Core is based on the Cognesys core engine, which provides natural language processing facilities and a cognitive system. The engine is customized to process project specific events. Communication between the two components is realized by a XML-RPC based protocol.

2.2 Interfaces

The interfaces to the other robot components are implemented as communication adapters that are part of the DM Communicator. This section lists all implemented interfaces with a short description of the used protocol and some examples.

Speech Recognition Interface (ASR)

Communication between the DM Communicator and the ASR component is based on HTTP. The ASR component sends recognized speech input as JSON arrays encapsulated in a HTTP POST (cf. RFC 2616) request. As the ASR component has two channels for speech recognition, the DM Communicator provides two URLs, one for each input channel:

- http://localhost/process-input-1 for the first input channel
- http://localhost/process-input-2 for the second input channel

The JSON array consists of several sequences, each representing a possible spoken user input. Each sequence is ordered by confidence. Listing 2.1 shows an example JSON array.

Listing 2.1: "ASR JSON array"

```
[
[["hallo", 1.0],["alle", 0.934]],
[["hallo", 1.0],["du", 0.76],["auch", 0.86]],
[["hallo", 1.0],["OOV", 0.23]]
]
```

In the example, three possible variants of recognized spoken input are provided. Each text is by itself a JSON array with tuples of words and the corresponding confidence value between 0 and 1. Not recognized words are represented by `OOV`.

Text-To-Speech Interface (TTS)

The robots TTS component implements a HTTP server that allows the DM Communicator to send XML-RPC request in order to provide spoken output to the user. The service is listening on http://localhost/speechserver and provides the method `ALIAS.TTS`. The method accepts an XML-RPC struct with two parameters: `command` and `argument`. To check whether the TTS component is ready to accept speech requests, the command `CMDstatus` with an empty argument is used. To perform speech output in German and French, the commands `TTSgerman` respectively `TTSfrench` are available. Both accept a string as argument.

ALIAS GUI Interface

The communication between the ALIAS GUI and the DM Communicator uses XML structs encoded in UTF-8. The packets are exchanged through UDP. There are three different types of packets: `command`, `request` and `signal`. Table 2.1 shows the different communication patterns between the DM Communicator and the ALIAS GUI. For example, as shown in the first pattern, when the DM receives speech input from the ASR, the NLP engine recognizes that the user requests to open the web browser. The DM chooses the action and sends corresponding command packet to the GUI. After starting the web browser, the GUI acknowledges the request with a signal packet.

Commands are sent from DM Communicator to the ALIAS GUI to initiate actions, e.g. starting a game or changing the menu. Signals are send from the ALIAS GUI to DM Communicator to inform the DM Core about state changes and events inside the GUI. A request packet is send from the ALIAS GUI to DM Communicator to request a command from DM Communicator. If the cognitive system inside the DM decides that the requested action can be executed by the GUI, the DM Communicator returns a corresponding command packet to the ALIAS GUI. Currently, request packets are not used. Each of the three packet types accepts the attributes `module`, `id` and `value`.

Table 2.2 shows an example communication during a skype call. First, the DM Communicator sends a command packet to start skype inside the GUI. The GUI answers with two packets, the first signal indicates that the GUI has received the skype command and the second confirms the successful start of the skype application. Then the DM Communicator sends another command packet to start the call with the desired phone number in

Table 2.1: GUI communication pattern

Command from DM, e.g. GUI control via speech input

Sender	Recipient	Packet Type
DM	GUI	command
GUI	DM	signal

Command inside the GUI, e.g. menu change

Sender	Recipient	Packet Type
GUI	GUI	command
GUI	DM	signal

Command from GUI via DM, e.g. program start

Sender	Recipient	Packet Type
GUI	DM	request
DM	GUI	command
GUI	DM	signal

Signal from GUI, e.g. incoming skype call

Sender	Recipient	Packet Type
GUI	DM	signal

the value attribute. Again, the GUI replies with a signal packet indicating that the phone connection has been initiated successfully. After the conversation, the DM communicator sends commands to close the call, to close skype and to go back to the main menu. In turn, the GUI acknowledges each command packet with a corresponding signal packet.

Robot Daemon Interface (RD)

The Robot Daemon (RD) on the SCITOS PC is responsible for the movement of the robot. To ensure a consistent robot behaviour, the RD and DM Communicator exchange information regarding the state of the DM Core and the state machine inside the RD. The RD provides a HTTP service with an XML based protocol. When the DM Communicator connects to the RD, it registers itself for events regarding the movement and the face identification of the robot. Table 2.3 shows the available events.

To control the movements of the robot, the DM Communicator sends `FIRE_EVENT` packets to the RD. As a confirmation, the RD responds by sending `FIRED` packets. When the robot has finished its moving, the RD sends an `EVENT` packet to the DM Communicator which indicates whether the robot could reach its target or not.

Table 2.4 shows an example. First, during robot startup, the DM Communicator registers the events. In the second step, the DM communicator informs the RD that the robot is

Table 2.2: DM communication with ALIAS GUI during a skype call

Sender	Packet
DM	<command module="app" id="id_skype"/>
GUI	<signal module="app" id="id_skype" value="ok"/>
GUI	<signal module="id_skype" id="start" value="ok"/>
DM	<command module="id_skype" id="call_by_number" value="+49123..."/>
GUI	<signal module="id_skype" id="call_by_number" value="ok"/>

... conversation ...

DM	<command module="id_skype" id="call_disconnect"/>
GUI	<signal module="id_skype" id="call_disconnect" value="ok"/>
DM	<command module="id_skype" id="close"/>
GUI	<signal module="id_skype" id="close" value="ok"/>
DM	<command module="menu" id="id_mainmenu"/>
GUI	<signal module="menu" id="id_mainmenu" value="ok"/>

Table 2.3: RD events

Event	Description
NavigatorGoalNotReachableEvent	RD was unable to reach the desired target
NavigatorGoalReachedEvent	RD has successfully reached its target
NavigatorPathPlannedEvent	RD has planned a path to reach its target
NavigatorNoValidDrivingCommandEvent	RD has received a invalid navigation command
FaceIDResult	RD has detected a face

in idle state. When the user orders the robot to approach him by using voice control, the DM Communicator fires an event, as indicated in step 3, in oder to move the robot to the user. Step 4 indicates that the RD has received the fired event and step 5 indicates that the robot has reached its target.

Brain Computer Interface (BCI)

Both the DM Communicator and the robots BCI component provide a UDP based service to exchange information via UTF-8 encoded XML. During startup, the BCI sends a broadcast packet, which allows listeners that receive this broadcast to register themselves as controllers for the BCI. The registering process is realized through a handshake procedure as described in the BCI manual [12]. The DM Communicator acts as such a controller by sending the BCI XML documents that define the application specific masks. The BCI presents these masks to the user and sends the corresponding user input back to the DM Communicator. Depending on the action chosen by the user, the DM Communicator sends another mask to the BCI or performs other actions. For example, if the BCI

Table 2.4: DM communication with RD

	Sender	Packet
1.	DM	<REGISTER_EVENT event="NavigatorGoalNotReachableEvent"/> <REGISTER_EVENT event="NavigatorGoalReachedEvent"/> <REGISTER_EVENT event="NavigatorPathPlannedEvent"/> <REGISTER_EVENT event="NavigatorNoPathPlannableEvent"/> <REGISTER_EVENT event="NavigatorNoValidDrivingCommandEvent"/> <REGISTER_EVENT event="FaceIDResult"/>
2.	DM	<FIRE_EVENT event="ModeChangeIdleEvent"/>
3.	DM	<FIRE_EVENT event="ModeChangeApproachEvent"/>
4.	RD	<EVENT event="NavigatorPathPlannedEvent"/>
5.	RD	<EVENT event="NavigatorGoalReachedEvent"/>

is in the main mask and the user chooses to start a skype call, the BCI sends this chosen action to the DM Communicator which in turn sends the mask to control skype back to the BCI. The BCI then displays this mask to the user in order to control the skype application. The implementation of a main mask and masks to control skype and audiobooks is still in progress.

Listing 2.2 shows the XML document describing the BCI main mask. This mask is initially transmitted from DM Communicator to the BCI component during the handshake procedure.

Listing 2.2: BCI main mask

```
<?xml version="1.0" encoding="utf-8"?>
<AppList xmlns:xsi ="http://www.w3.org/2001/XMLSchema-instance" xsi:
    noNamespaceSchemaLocation="XMLSchema_UDPinterface_v3.xsd">
  <ControlGroup CGName="ALIAS_BCI_CONTROL">
    <CGAddress>
      <IPAddress>134.106.140.149</IPAddress> <!-- set apropriate ip adress -->
      <Port>22346</Port> <!-- define appropriate port see manual -->
    </CGAddress>
    <MaskSize>
      <NoLines>3</NoLines>
      <NoCols>3</NoCols>
    </MaskSize>
    <DispSymbol>
      <Text>ALIAS</Text>
    </DispSymbol>
    <SamplingRate>-1</SamplingRate>
    <Application AppName="MainMenu">
      <AppID>AB</AppID>
        <SingleCommand CommandName="StartSkype"><!-- insert audiobookname of contact-->
          <Instruction>run</Instruction>
          <ICONPosition>[1,1]</ICONPosition>
          <DispSymbol>
            <Text>Skype</Text> <!-- insert audiobookname or nick of person to contact
                -->
          </DispSymbol>
          <CommType>multi</CommType>
          <CommandParameter>
            <!-- fill in apropriate audiobook name of contact to be contacted -->
            <Parameter type="application">skype</Parameter>
          </CommandParameter>
        </SingleCommand>
        <SingleCommand CommandName="StartAudioBooks">
          <!-- insert audiobookname of contact-->
          <Instruction>run</Instruction>
          <ICONPosition>[2,1]</ICONPosition>
          <DispSymbol>
            <Text>Books</Text>
            <!-- insert audiobookname or nick of person to contact -->
          </DispSymbol>
          <CommType>multi</CommType>
          <CommandParameter>
            <!-- fill in apropriate audiobook name of contact to be contacted -->
            <Parameter type="application">audiobooks</Parameter>
          </CommandParameter>
        </SingleCommand>
        <SingleCommand CommandName="BrowseTheWeb">
          <!-- insert audiobookname of contact-->
          <Instruction>run</Instruction>
          <ICONPosition>[2,3]</ICONPosition>
          <DispSymbol>
            <Text>www</Text>
            <!-- insert audiobookname or nick of person to contact -->
          </DispSymbol>
          <CommType>multi</CommType>
          <CommandParameter>
            <!-- fill in apropriate audiobook name of contact to be contacted -->
            <Parameter type="application">webbrowser</Parameter>
          </CommandParameter>
        </SingleCommand>
    </Application>
    </ControlGroup>
 </AppList>
```

Listing 2.3 shows a message send from BCI to the DM Communicator during a skype chat. The user entered the word HALLO and sent the message.

Listing 2.3: BCI message during a skype chat

```
<?xml version="1.0" encoding="utf-8">
<command>
<ID>CM_BCI_001</ID>
<AppID>SP_RC_001</AppID>
<Instruction>send</Instruction>
<parameter type="message">HALLO</parameter>
</command>
```

Further details on the used XML protocol are available in the BCI Manual [12].

3 Person Detection

In order to approach a user to advance to a human-machine dialogue, a module for person detection is needed to determine the position of the user. For the ALIAS system a module is applied that combines face detection with laser-based leg-pair detection. Both face and leg-pair detection are running in parallel and their output is combined to create a more stable and robust person detection hypothesis.

3.1 Camera-based Face Detection

The face detection system has been implemented for the face identification system of the ALIAS platform [14]. It uses the omni-directional camera that is mounted on top of the robotic head. This camera provides a 360-degree image with a resolution of 720×280 pixels that enables the detection of faces in all directions of the robot. To detect faces, the algorithm introduced by Viola and Jones [22] is used. This algorithm is characterized by its very effective and fast processing, and those allows high detection rates. The fast processing speed can be attributed to three properties: First, a so-called *integral image* is used to compute features, second adaptive boosting (adaBoost) [11] is applied and third, the cascade structure of classifiers speeds up the detection process.

Images recorded with the camera are first transformed to grey-scale images. Then, a so-called Haar cascade classifier is used to detect faces. Haar-like features are computed on the integral image and adaBoost is applied by using a cascade of weak classifiers for detecting the human face. The hypothesis with the highest probability is used, resulting in only one face being detected at a time.

Due to the hardware specifications of the camera, the face detection performance is limited in practice. Faces can be detected with frame rate of up to roughly $10 - 20Hz$ and up to a distance of $2 - 3m$. In a single-user scenario, face detection works very reliable. If multiple users are around the robot, mostly the person nearest to the camera will be detected.

3.2 Laser-Based Leg Pair Detection

The second person detection channel is the leg pair detector on the robot. We use a SICK S300 laser range finder that is able to measure the distance of obstacles within 270 degrees around the robot. Here, this scanner is able to scan a two dimensional slice of the robot's three dimensional operation environment only (see figure 3.1 for an example). The angular resolution is 0.5 degree and measurements could be made up to 30 meters with a error tolerance of 5mm.

Each scan results in a series of distance measurements, each corresponding to one scanning angle. The goal of the used method is to find a subsection of points, which corre-

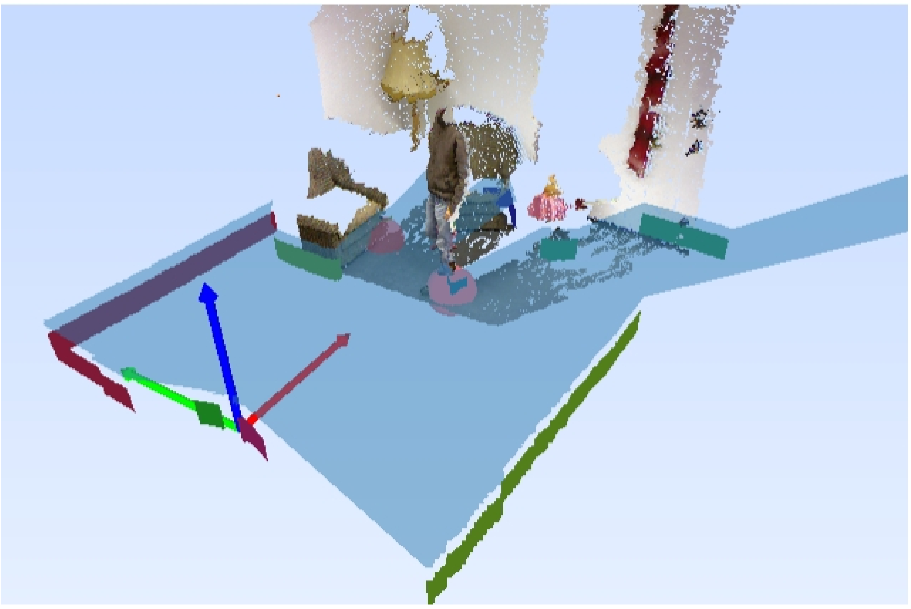


Figure 3.1: Example of the laser based leg detection. The laser is able to detect obstacles within 270 degrees in front of the robot (blue fan). Each scan is segmented into multiple segments (colored rim). For each segment a set of ten features is calculated and afterwards classified into legs and non-legs.

spond to person legs. To do so, the method of Arras [2] is used. First, the scan is clustered into different segments by using an incremental segmentation. Here, the main idea is, that the last point of a segment is compared with the successor. If the distance is above a defined threshold, a new segment is created, and if not, the successor belongs to the current segment.

Up to this point, the scan is split into a set of segments. Then, for each segment a set of descriptive features is calculated that describe the segment's size, curvature, roughness and others. For detailed information, please refer to [2]. Finally, ten descriptive features are calculated for each segment. These features are classified by a random forest classifier into the classes *leg* or *non-leg*. The random forest classifier is the main deviation from the original proposed method, which uses the AdaBoost method for classification. All segments that are classified as legs are filtered afterwards, so that nearby leg-like segment are fused towards one person hypothesis.

There are two software modules needed, to build a module for leg detection. On the one hand, one module has to collect positive and negative examples of leg segments and non-leg segments. This module operates only on a static robot and learns a background model of the environment. Each segment that does not belong to the background is than recorded as a positive example of a leg and those a training set of positive examples is constructed. The negative examples where recorded with the same module on a driving robot without a background model. Here, every segment is seen as a example of a non-leg. During the recording the environment should not contain persons. The collection of training data only has to be done in the training phase of the system (usually once in a lifetime) with the set of positive and negative leg examples.

In the operation phase of this method, which is within the second software module, the trained classifier is used. This module is designed as a blackboard client to operate in the robot system during regular operation. Here, the same preprocessing of segmentation and feature extraction is done. The classification per scan with multiple segments is very fast in terms of computing time and ca be done in 1-3 ms on the available robot hardware.

Although this approach is very fast, it can detect a lot leg-like objects within a home environment. This is mainly due to the similar shape of a 2D cut through three dimensional objects. Garbage bins, chairs, table legs and window frames look very similar to legs, for instance, and are thus also detected. this means that this classifier has a very high true-positive rate, but at the same time a high false-positive rate.

To conclude this section, the presented approach is able to detect leg pairs, if they are within the laser scan, which works fine up to 5-8 meters (which is sufficient for home environments). However a drawback is that a lot of leg-hypotheses are made that are actually no legs. To improve the reliability and robustness of the face and leg-detection approach, both hypotheses channels are fused.

3.3 Fusion

In order to increase person detection robustness and combine the advantages of camera-based face detection and laser-based leg detection, both systems are fused to produce a combined detection hypothesis. Since both channels give robot-centered metric information, about the position of a person in terms of angle and distance, both channels can be fused. The simple idea is to assign a reliability value to the hypotheses. To find corresponding hypotheses in both channels, the pair wise distances are evaluated. Those hypotheses that which are very close to each other will get higher reliability values than those with huge distances. Hypotheses without any correspondence are rated lower. In such cases, a hypothesis from the laser channel is rated lower (because of the high false-error rate) than a hypothesis from the camera channel, which has a very low false positive rate. Only the set of the most likely hypotheses are used for the dialogue system.

4 Speaker Identification

Speech signals do not only carry linguistic but also paralinguistic information enabling to estimate speaker's identity. Automatic speaker recognition systems attempt to identify a person based on his/her voice. Such systems can be used either to determine the speaker's identity or to verify the claimed identity of a person. Speaker identification is an essential step towards personalizing spoken dialogue systems in order to make human-machine interactions as natural as possible and is hence relevant from the ALIAS project perspective.

One particular advantage in using speech for identifying users of the ALIAS robot is the ease with which the signal can be obtained. Speech signals are readily captured in almost any environment using standard microphones and recording equipment and do not depend on the orientation of a camera or relative position of the subject. It is further independent of occlusion and inter-session variations in illumination, pose or expression which often degrade the performance of face recognition systems in similarly uncontrolled contexts. Speaker recognition, however, is not without its own specific issues related to inter-session variation. Ambient noise, differences in the linguistic context, a persons state of health or emotional state all influence performance. The quantity of data is also an important factor. Whereas face recognition may only require a single image, speech signals are dynamic, i.e. information is contained within its variation over time. Sufficient data is thus required for acceptable performance and place certain constraints on viable applications and contexts relevant to the ALIAS project.

In the following sections, the speaker identification module as it is implemented on the ALIAS robot is described in detail.

4.1 Speaker identification system

The Speaker identification module implemented on the ALIAS Robot is based on ALIZE/LIA_RAL [4] system using SPRO [15] for front end feature extraction. The block diagram of the speaker identification module is shown in figure 4.1. The system is based on Gaussian mixture models (GMMs) using a universal background model (UBM) paradigm [19].

4.1.1 Feature Extraction

The speech signal is transformed to linear frequency cepstral coefficients (LFCCs) using a FFT window size of 20ms and a frame shift of 10ms. The features comprise 16 cepstral coefficients along with energy appended with delta and delta-energy coefficients leading to 34 dimensional vectors. The speech signal is band limited to 300-3400 Hz during feature extraction.

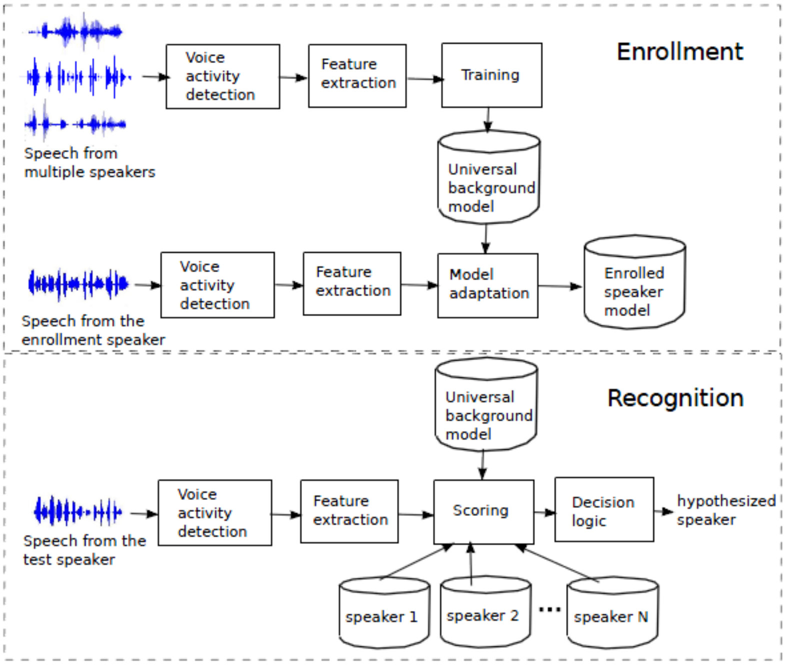


Figure 4.1: Block diagram of speaker identification module

4.1.2 Voice Activity Detection

This step is for the removal of non-speech frames from further processing. For this, the energy coefficients are first normalised to zero mean and unit variance. The normalised features are then used to train a GMM with 3 mixture components that is used to screen out the frames with low energy corresponding to the non-speech signal. After the silence frame removal, the remaining frames are again normalised to zero mean and unit variance.

4.1.3 Universal background model (UBM)

Speech data from several speakers is pooled to train a background GMM using the expectation-maximisation algorithm [10]. For the ALIAS system, due to the absence of a large volume of training data, the UBM was trained on the CHiME corpus [9]. Since the computational time required to identify a speaker is important due to its real time nature on the ALIAS robot, the number of Gaussian components in the UBM has been set to a relatively smaller value of 64.

4.1.4 Speaker model generation

Each speaker is modeled by using a single GMM. Due to the limited amount of data available for enrolling each speaker, individual speaker models are obtained by adapting the UBM with the enrollment data using maximum-a-posteriori approach [13].

4.2 Audio path

The sound capture devices on the ALIAS robot i.e., either 'Audio 4 DJ' or 'Traktor Audio 6' are connected to the windows system as per the system design, while the speaker identification module runs on the linux system as a blackboard client. Hence an audio path to the input of speaker identification module has been setup as shown in the figure 4.2.

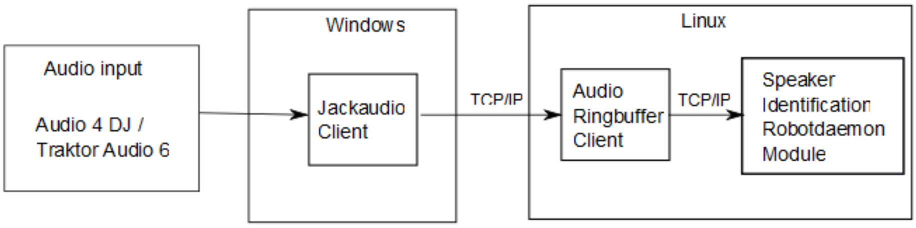


Figure 4.2: Audio Path for the speaker identification module on the ALIAS Robot

Upon request, a JackAudio client on the windows machine streams the audio on a TCP/IP connection. An audio client on the linux machine connects to this Jack client over the ethernet connection and continuously receives the audio stream. This stream is stored in a reconfigurable ring buffer and currently set to hold 30 seconds. The speaker identification module in turn connects to this ring buffer client when required over a TCP/IP connection to receive either the recent past or current few seconds of audio data.

4.3 Robotdaemon Application

As mentioned above, the speaker identification module has been setup as a blackboard client and is loaded upon execution of the Robotdaemon application . The configuration parameters related to the client such as TCP/IP settings to connect to the ring buffer, audio channel details and audio duration for enrollment and testing can be configured by editing the file './clients/SpkDetClient_config.xml' in the Robotdaemon application folder. Configuration files specific to the implementation of the submodules of the Speaker identification module can be found in the directory './SpkDetClient/configs'. The client offers two methods, one for speaker enrollment and one for identifying the speaker as described below.

4.4 Enrollment and Testing

In order to enroll a new speaker, the function trainTargetSpkDet(SpkName) needs to be invoked with the speaker name as the input argument. This will record audio with duration as set in the configuration file, extract features, perform voice activity detection, normalise the features and adapt the UBM and store the new speaker model in the folder './SpkDetClient/mixtures/test/' as 'SpkName.gmm'.

In order to recognise the identity of a speaker, the function recogniseSpeakerSpkDet() needs to be invoked. This function operates in two different modes

1. Identify the speaker from the current audio being captured. This functionality can be obtained by setting the input argument to the function to '1' i.e., recogniseSpeakerSpkDet(1).
2. Identify the speaker from the past few seconds of audio by setting input argument to '-1' i.e., recogniseSpeakerSpkDet(-1). This mode is particularly useful as most of the times the speaker needs to be identified after he/she has spoken.

The speaker identification function computes the likelihood score of all the existing speaker models with respect to the test utterance and returns the speaker name with the highest score.

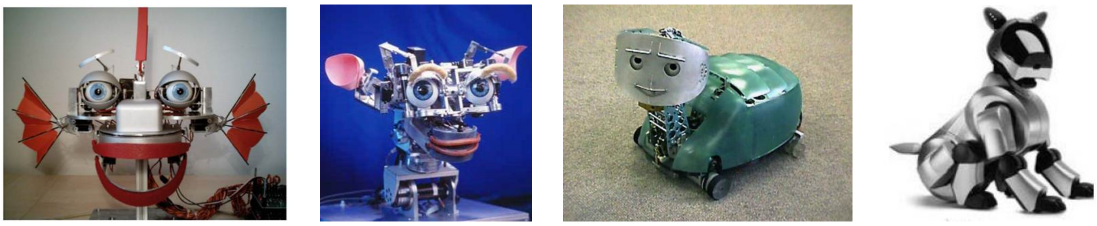
5 Display of Emotions with ALIAS

In this chapter, we describe how the robotic head of the robotic platform ALIAS is used to display emotions. Emotions are an important communication channel and therefore, the human-robot dialogue can be enriched by displaying emotions. ALIAS can thus better adapt to its users during a dialogue. The robotic head has several degrees of freedom to display different facial expressions. Five different facial expressions corresponding to five different emotions have been developed. However, due to the absence of a mouth and eye brows, it is especially difficult to display emotions that can be identified by humans. In an experimental section, we show how good humans can recognise the displayed emotions.

5.1 Introduction

Emotions are an important aspect of human-human communication. Based on the work of Freud, Zimbardo and Ruch describe communication in an ice berg model [23]. Thereby, only 20 % of communication consists of a visible part, which involves facts and figures. The larger part consists of non-visible aspects like personality, fears, conflicts and emotions. In communication, emotions have the functions of dialogue control, transmission of information, social bonding, competence and personalisation. This model concerns mainly human-human communication. But since it is desired to make human-machine interaction as natural as possible, it is necessary to include emotions in a human-machine dialogue. While industrial robots are not well-suited for the display of emotions, especially robotic platforms in the fields of health care, entertainment or service robotics can be enriched by emotions.

Several robots with abilities to display emotions have been developed. In Figure 5.1, four examples for robotic platforms that can display emotions are given.



(a) EDDIE. From: [21] (b) Kismet. From: [6] (c) Sparky. From: [20] (d) AIBO. From: [1]

Figure 5.1: Different examples of robots capable of displaying emotions

EDDIE (Emotion Display with Dynamic Intuitive Expressions) [21], displayed in figure 5.1a is a human-like robotic head. It has 23 degrees of freedom at its eyes, eye brows, ears, mouth and jaw. In addition, two animal-like features are mounted in order to strengthen the ability to display emotions. EDDIE is also developed with child-like characteristics (e. g. the large eyes). The basic emotions joy, surprise, anger, disgust, sadness and fear can be displayed with EDDIE.

In [3], a robotic platform based on EDDIE that enriches a multimodal human-robot dialogue with emotional feedback was described.

Similar to EDDIE, the robotic head Kismet [6] resembles a human head, but without any additional animal-like features. Kismet has 15 degrees of freedom at its eyes, eye brows and lids, neck, ears, lips and mouth. It can display the emotions happiness, sadness, surprise, anger, calm, displeasure, fear, interest and boredom.

Compared to the robotic heads described so far, Sparky [20] is relatively small with its dimensions of $60 \times 50 \times 35cm$ but it consists of a whole body and not only a head. With its cartoon-like character, the uncanney valley [17] can be dodged. In order to display emotions, Sparky has only ten degrees of freedom: Its eye brows and lids, top and bottom lip, neck, back plate and wheels are movable. The tiltable back plate can be interpreted as an animal-like feature. Sparky can display the emotions happiness, sadness, anger, surprise, fear, curiosity, nervousness and sleepyness.

The robot dog AIBO [1] (Artificial Intelligence roBOt) is an already commercially available robotic platform. It has 20 actuators and can move its mouth, head, ears, tail and legs. AIBO displays not only separate emotions, but complete behaviours, which are exploring, demanding and giving attention, fear, playing, learning and seeking protection.

Further examples for social robots capable of displaying emotions are CERO [16], FEELIX [8], VIKIA [7], PARO or the Sony Dream Robot.

All of the presented robotic platforms have different features to display emotions, where especially the mouth, eyes and eye brows play an important role.

The robot ALIAS is designed as a communication platform for elderly persons [18]. Therefore it is well suited for the incorporation of the ability to display emotions, to enrich the human-machine dialogue. Since it has no mouth, it is especially difficult to display emotions with ALIAS. Five different emotional facial expressions have been developed and evaluated with the robotic platform ALIAS.

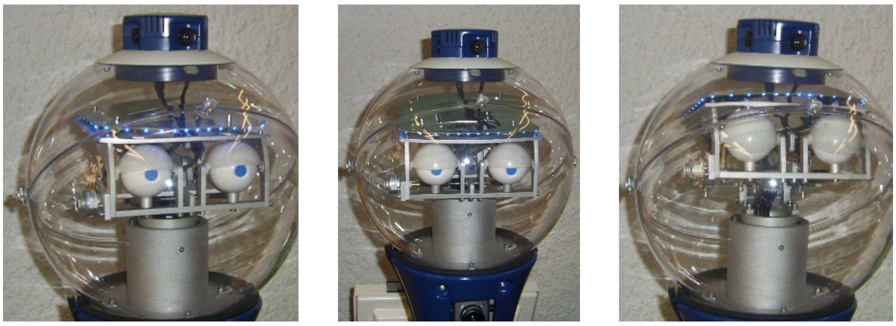
In the next section, we present the hardware of the head of the robotic platform ALIAS. The implemented emotional facial expressions are described in Section 5.3. Experimental results are presented in Section 5.4 before conclusions are given in the last section.

5.2 Robotic platform ALIAS

The robotic platform ALIAS is based on the platform SCITOS-G5 of MetraLabs. It is equipped with a head which is used to display emotions. The head has several degrees of freedom. It can be turned 360 degrees horizontally and vertically up (15 degrees) and down (6 degrees). The two synchronized movable eyes can be turned up to 8.5 degrees in both horizontal directions and the eye lids can be opened or closed on a continuous scale. Horizontal rotation of head and eyes is not used to distinguish between emotions, in order to ensure the possibility to perceive emotions independent of the viewing angle. Above the eyes, a row of blue LEDs is mounted, which can be turned on and off or set running or blinking with any frequency. In addition, the brightness of the LEDs can be controlled. In total, this sums up to 6 degrees of freedom that are used to display different facial expressions. Compared to other robotic heads, this is a relatively small number of degrees of freedom. It should also be noted that ALIAS has no mouth or eye brows, which are important for displaying emotions. Therefore, it will be rather difficult to display emotions with ALIAS.

5.3 Emotional facial expressions

In addition to a neutral facial expression, five emotional facial expressions have been implemented. The neutral facial expression is displayed in Figure 5.2a and is characterized by slightly closed eyes and unmoved head while the LEDs are *on* with a medium brightness. In Figure 5.2b, the facial expression showing sadness is displayed. The head is turned down with slightly closed eyes and the LEDs are blinking with a slow frequency. The facial expression for the disgust emotion is displayed in Figure 5.2c. The robotic eyes are completely closed and the head looks upwards.



(a) Neutral facial expression (b) Sad facial expression (c) Disgust facial expression

Figure 5.2: Different facial expressions of the robotic head of the ALIAS platform

In order to display happiness, the LEDs are set *running* and for fear, the eyes are wide open and the LEDs are set *blinking* with a high freqency. Surprise is displayed by LEDs blinking with a very high frequency.

In Table 5.1, all settings of the head for the implemented emotions are displayed.

5.4 Experiments

A preliminary user study has been conducted to evaluate how good humans can identify the emotions displayed by ALIAS. 12 male adult subjects with ages between 23 and 30 participated in the study. The robot was presented to the users and the five different emotions have been shown five times each, in a random order. Subjects had to identify the

Emotion	Head tilt	Eye lid	LEDs	LED frequency	LED brightness
Neutral	0.0	0.85	on	-	0.5
Sadness	-6.5	0.81	blinking	220	0.2
Disgust	12.5	0.0	on	-	0.2
Fear	0.0	1.0	blinking	35	1
Surprise	0.0	1.0	blinking	10	1
Happiness	0.0	1.0	running	35	1

Table 5.1: Settings for the actuators of the robotic head of ALIAS for the implemented emotions

displayed emotion and could choose from a list of 10 different emotions. The 10 possible answers are the same Breazeal used to evaluate the robot Kismet [5]. In addition, subjects rated their guess on a scale from 1 (very unsure) to 10 (very sure). Table 5.2 shows the identification rates resulting from the user study.

Emotion	Identification rate (in %)
Sadness	38.8
Disgust	33.3
Fear	6.6
Surprise	33.3
Happiness	23.3
Mean	27.1

Table 5.2: Identification rates for emotions displayed with the robotic head of ALIAS

Overall, the identification rates are very low. Sadness achieves the highest accuracy with 38.8 % while fear was only identified 6.6% of the time. However, there are clear tendencies visible in the confusions of different emotions: Disgust and sadness have been reconised as boredom very often, joy as curiosity and surprise as anger. This can mainly be attributed to the absence of a mouth, which is a very important feature for display of emotions. On the other hand, the LEDs are the main feature that is used to differentiate between emotions, supported by head tilt and eye lids.

5.5 Conclusions

It was evaluated how different emotional facial expressions could be recognised in a user study. While the identification rates for the emotions are very low, it is already very promising that with such limited hardware it is possible at all to distinguish between different emotions. The addition of a mouth could be very promising to increase the identification rates of emotions, as well as the introduction of different colours for the

LEDs. Reducing the set of different emotions could also lead a to better separation by the human observer.

6 Use Case Scenarios

This chapter describes the prepared use-case scenarios of ALIAS and how the dialogue system is involved in the process of the scenario. Thus, it can be shown how the dialogue manager works in practice.

6.1 Alarm Call Scenario

In this scenario, many of the functionalities of ALIAS are included: Speech in- and output, autonomous navigation, video telephone functionality, remote control of the robot and further entertainment functionalities. Speech commands are used to trigger an alarm call. ALIAS then drives to his observation position and establishes a video telephone connection to a caregiver. The caregiver can then remote control the robot to drive near the user and carry out a video call. With the robot in reach, the subject can use the graphical user interface to play one of the games or browse in the internet. Table 6.1 describes sequence of actions and the involved robot components in the alarm call scenario.

6.2 BCI Scenario

This scenario displays the usage of a Brain-Computer Interface (BCI) in conjunction with the robotic platform ALIAS. The BCI can be used to control the Graphical User Interface(GUI) of the robot and to start the playback of an audiobook. By using the BCI as an input modality, the user can navigate through the menu to select an audiobook. The playback of the audiobook can be started and stopped with the BCI as well.

As the BCI is an alternative way to control the GUI, the DM acts as a translator between the two components. Based on the user input that is sent from the BCI to the DM Communicator, the DM updates the BCI masks or sends corresponding commands to the GUI. The scenario starts in the main mask of the BCI. The user first selects the audiobooks sub mask. The BCI sends a corresponding packet to the DM and the DM decides to send the audiobooks mask back to the BCI and open the audiobooks menu in the GUI. Then, the users starts a audiobook by selecting the corresponding mask entry in the BCI. Respectively the BCI again sends a corresponding packet to the DM. The DM sends a command to the GUI to start the audiobook and a new mask to the BCI to control the audiobook playback.

Table 6.1: Alarm call scenario

	Involved Components	Description
1.	ASR, DM	The user says “Robin Hilfe!”. The ASR sends the recognized text to the DM Communicator. DM Communicator forwards the text to DM Core. DM Core decides to start the alarm scenario.
2.	DM, TTS, RD	In order to navigate to the user, the DM fires an event to the RD and uses the TTS service to answer “Ich komme!”.
3.	RD, DM	The RD navigates the robot to the user and then fires an event to inform the DM that he has approached the user.
4.	DM, TTS, GUI	The DM uses the GUI to start a ten second alarm countdown and uses the TTS to say “Der Notruf wird in zehn Sekunden ausgelöst. Bitte sagen sie Stop, wenn sie keine Hilfe benötigen”.
5.	GUI, DM, TTS	After the countdown has finished, the GUI informs the DM. The DM decides to start a skype video call to the doctor, sends a corresponding command to the GUI and uses the TTS to say “Eine Telefonverbindung wird hergestellt”.
6.	DM	While the user and the doctor have their video call, the DM ignores text input from ASR.
7.	RD, GUI	The doctor uses the manual navigation to approach the user in order to check the users health. The doctor advises the user to play a game to calm down. They conclude the video call.
8.	GUI, DM	The GUI informs the DM that the skype call has finished. The DM decides to stop ignoring the ASR.
9.	ASR, DM, GUI, TTS	In order to start a game, the user says “Hauptmenü”. The ASR sends the recognized speech input as text to the DM. The DM decides to open the main menu and sends a corresponding command to the GUI. Furthermore, the DM uses the TTS to say “Hauptmenü”. The GUI opens the main menu. Then, the user says “Welche Spiele hast Du?”. Again, the ASR sends the text to the DM, the DM decides to open the game menu and uses the TTS to say: “Ich öffne die Liste der Spiele.”. Then the users says “Bitte starte Solitär”. Again, the ASR sends the text to the DM, which in turn starts the game solitaire and uses the TTS to say “Ich starte das Spiel Solitär”.
10.	ASR, DM, GUI, TTS	After finishing the game, the user again uses speech input to return to the main menu and start the internet browser. The robots components react in the same way as in step 9.

7 Conclusions

In this document, the latest version of the dialogue manager and some connected modules are described. The architecture of the dialogue manager as well as the definition of interfaces to most of the other robot modules is addressed. In its current form, the dialogue manager communicates with all other modules which are used on the robotic platform.

An approach based on face detection and leg pair detection is used to detect persons. The fusion of these two technologies leads to a robust and stable detection of users. After a person has been detected, speaker identification can be used to identify the person. The employed speaker identification system has been described in this document. Finally, it is described how ALIAS displays emotions in order to give natural feedback to the users.

Furthermore, two use case scenarios are described in order to show how the dialogue system and other modules are applied.

The dialogue manager and its connected multimodal input and output modules lead to a natural human-robot dialogue.

Bibliography

[1] R. Arkin. An ethological and emotional basis for human–robot interaction. *Robotics and Autonomous Systems*, 42(3-4):191–201, 2003.

[2] K. Arras, O. Mozos, and W. Burgard. Using boosted features for the detection of people in 2d range data. In *Proc. ICRA*, 2007.

[3] A. Bannat, J. Blume, J. Geiger, T. Rehrl, F. Wallhoff, C. Mayer, B. Radig, S. Sosnowski, and K. Kühnlenz. A multimodal human-robot-dialog applying emotional feedbacks. In *Proc. Social Robotics: Second Intern. Conf. on Social Robotics, ICSR 2010, Singapore*, number LNAI 6414, pages 1–10. Springer, Berlin Heidelberg, 2010. 23.-24.11.2010.

[4] J.-F. Bonastre, F. Wils, and S. Meignier. ALIZE, a free toolkit for speaker recognition. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, volume 1, pages 737 – 740, 18-23, 2005.

[5] C. Breazeal. *Designing sociable robots.* The MIT Press, 2004.

[6] C. Breazeal and B. Scassellati. How to build robots that make friends and influence people. *Intelligent Robots and Systems, 1999. IROS '99. Proceedings. 1999 IEEE/RSJ International Conference on*, (2):858–863, 1999.

[7] A. Bruce, I. Nourbakhsh, and R. Simmons. The role of expressiveness and attention in human-robot interaction. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 4, pages 4138–4142. IEEE, 2002.

[8] L. Cañamero. Playing the emotion game with feelix. *Socially Intelligent Agents*, pages 69–76, 2002.

[9] H. Christensen, J. Barker, N. Ma, and P. Green. The CHiME corpus: A resource and a challenge for computational hearing in multisource environments. In *Interspeech*, 2010.

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[11] Y. Freund and R. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer, 1995.

[12] g-tec Medical Engineering. *BCI XML Standard Manual v1.0.* g-tec Medical Engineering.

[13] J.-L. Gauvain and C.-H. Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE Transactions on Speech and Audio Processing*, 2:291–298, 1994.

[14] J. Geiger, T. Rehrl, R. Vipperla, and N. Evans. D3.3 - Documented Identification System Basing on Face and Voice. ALIAS Deliverable, 2011.

[15] G. Gravier. SPRO : a free speech signal processing toolkit, 2004.

[16] H. Huttenrauch and K. Eklundh. Fetch-and-carry with cero: observations from a long-term user study with a service robot. In *Robot and Human Interactive Communication, 2002. Proceedings. 11th IEEE International Workshop on*, pages 158–163. IEEE, 2002.

[17] M. Mori. The Uncanny Valley. *Energy*, 7(4):33–35, 1970.

[18] T. Rehrl, J. Blume, J. Geiger, A. Bannat, F. Wallhoff, S. Ihsen, Y. Jeanrenaud, M. Merten, B. Schönebeck, S. Glende, and C. Nedopil. Alias: Der anpassungsfähige ambient living assistent. In *Tagungsband des 4. Deutschen Ambient Assisted Living (AAL 2011) Kongresses, Berlin*, page 9 Seiten. VDE Verlag, 2011. 25.-26.01.2011, ISBN 978-3-8007-3323-1.

[19] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10(1-3):19–41, 2000.

[20] M. Scheeff, J. Pinto, K. Rahardja, S. Snibbe, and R. Tow. Experiences with Sparky, a Social Robot. *Multiagent Systems, Artificial Societies, and Simulated Organizations*, 3:173–180, 2002.

[21] S. Sosnowski, A. Bittermann, K. Kuhnlenz, and M. Buss. Design and Evaluation of Emotion-Display EDDIE. *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 3113–3118, 2006.

[22] P. Viola and M. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[23] P. G. Zimbardo and F. L. Ruch. *Lehrbuch der Psychologie: Eine Einführung für Studenten der Psychologie, Medizin und Pädagogik.* Springer, Berlin West, 3 edition, 1978.