AAL-2009-2-049, ALIAS
D3.8
Final Dialogue System

Due Date of Deliverable	2013-02-28
Actual Submission Date	2013-02-28
Workpackage:	3.8
Dissemination Level:	Public
Nature:	Report
Approval Status:	Final
Version:	v1.0
Total Number of Pages:	30
Filename:	D3.8-MMK-DM-v1.0.pdf
Keyword list:	Dialogue System, Dialogue Manager, Human-Machine Communication

Abstract

Following the series of deliverables describing the newest version of the dialogue system, this deliverable now describes the final dialogue system of the ALIAS robot and its involved components. To control the robot, several input modalities can be used: speech commands, a touch-screen, or a BCI. All input is processed by the dialogue manager, which decides about the next action to be taken. In this document, the involved hardware and software components are described. To demonstrate the functionality of the system, five use-case scenario have been developed and tested with elderly users.

History

Version	Date	Reason	RevisedBy
0.1	2013-02-05	created [MMK]	Jürgen Geiger
0.2	2013-02-19	reviewed [FhG]	Sven Fischer
0.3	2013-02-19	updated [MMK]	Jürgen Geiger
0.4	2013-02-25	reviewed [EUR]	Raphaël Troncy
1.0	2013-02-28	finalised [MMK]	Jürgen Geiger

Authors

Partner	Name	Phone / Fax / Email
MMK	Jürgen 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 4
2 System Overview 5
3 Hardware Components 8
4 Dialogue System Components 10
4.1 Dialogue Manager 10
4.1.1 Architecture 11
4.1.2 Interfaces 12
4.2 User Interface 17
4.2.1 Input Modules 17
4.2.2 Output Modules 18
4.2.3 Sensor Modules 18
4.3 Applications 20
4.3.1 Reader Functionality 20
4.3.2 Games 21
4.3.3 Audio Books 21
4.3.4 TV 21
4.3.5 Internet Browser 21
4.3.6 EventMedia 22
4.3.7 Telephone 22
4.3.8 Physiological Monitoring 22
4.3.9 Remote Control 23
5 Use-case scenarios 26
5.1 Self-experience scenario 26
5.2 Remote Control by Secondary User 26
5.3 BCI Scenario 26
5.4 Ground Lighting Scenario 27
5.5 Event scenario 27
6 Conclusions 28

1 Introduction

The central component of the ALIAS robot is the dialogue system. It is built around the dialogue manager, which is fed by input modules and controls other modules. The dialogue manager is the "brain" of the robot. Using input from a speech recogniser, a BCI system, or a GUI, the dialogue manager decides for the next action to be taken. Furthermore, input from various sensor modules is processed. To demonstrate how the dialogue system works together with the input and output modules and further applications, five use-case scenarios have been implemented.

In Chapter 2 an overview over the whole system including all involved hardware and software components is given. More details about the hardware is given in Chapter 3. The involved software components are described in Chapter 4. In Chapter 5, the implemented use-case scenarios are described. Finally, this Deliverable is concluded in Chapter 6.

2 System Overview

This chapter gives a short overview over all components involved in the dialogue system. This includes hardware components and software modules. The list of hardware components consists mainly of components used for user in- and output. Various software modules are used to process user input and forward it to the dialogue manager. The dialogue manager itself interpretes user input and decides for the next action to be taken. It can control several other software applications. Finally, there are various software modules to generate output for the user.

The robotic platform ALIAS is composed of several hardware units. Figure 2.1 shows the robot. The basis of the robot is the driving unit. It uses a differential drive system. A laser range finder, ultrasonic sensors and a collision sensor are used by the robot to localize itself, navigate autonomously and approach a user in a socially acceptable manner [8].

Above the driving unit, the interaction unit is mounted. The interaction unit consists mainly of a touch-screen and the robotic head. While the touch-screen is used for user input and output, the robot's head is used for additional feedback. Around the touch-screen, microphones are mounted to record speech. Additionally, loudspeakers are mounted below the touch-screen. The robotic head itself can be moved with several degrees of freedom. Different facial expressions can be displayed with the robotic head. Additionally, a row of LEDs is mounted in the upper part of it. Above the head, an omni-directional camera is mounted, which is used for face detection. In total, the robot is approximately $1.5\ m$ tall.

All software modules of the robotic system are centered around the dialogue manager. Figure 2.2 shows an overview over all software modules connected to the dialogue manager. The dialogue manager is the core of the dialogue system. It processes input from various other software modules and sends commands to other modules. Within the dialogue manager, a natural language understanding (NLU) unit processes the incoming information and decides for the next action to be taken. In particular, the dialogue manager controls all modules of the user interface. The primary input channel is speech: an automatic speech recognition (ASR) module is used to process speech commands recorded with a microphone. The result of the speech recognition module is forwarded to the dialogue manager, which interpretes it. The touch-screen is another important input channel. Using buttons on the graphical user interface (GUI), the user can directly control the available modules. Applications can directly be started with the touch-screen. On the other hand, the GUI is used to give feedback to the user. It displays the current state of the robot and the user can navigate through the menu of applications. Using a text-to-speech (TTS) module, speech output is generated by the dialogue manager as an answer to the speech input generated by the ASR system. Additional feedback is given to the user with the robotic head: Different emotional facial expressions are used to display the internal state of the robot. In addition, ALIAS tries to keep eye contact by moving its head and eyes

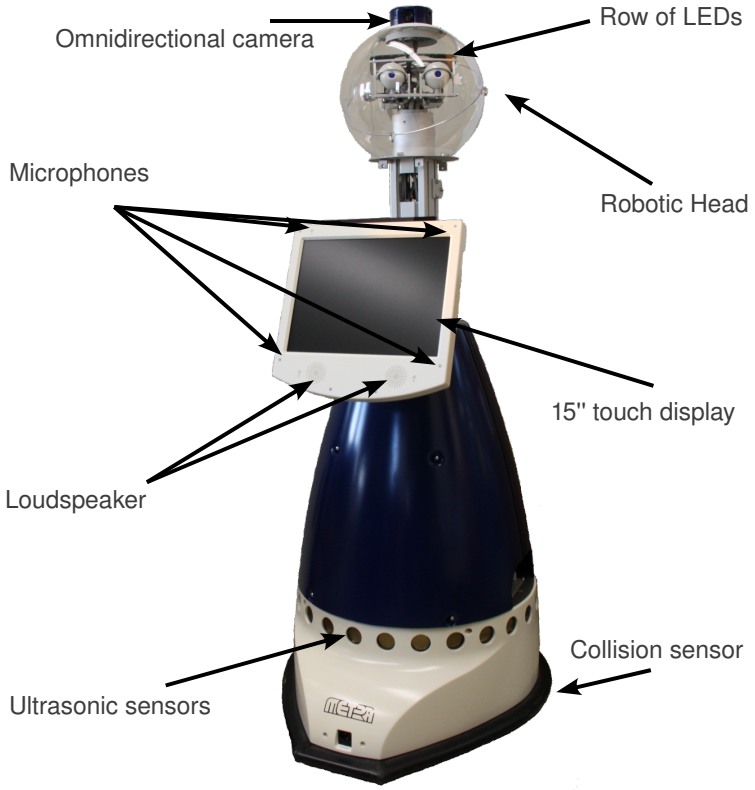


Figure 2.1: Overview of the hardware setup of the robotic platform ALIAS, divided into driving unit (lower part) and interaction unit (upper part) consisting of the touch-screen display and the robotic head).

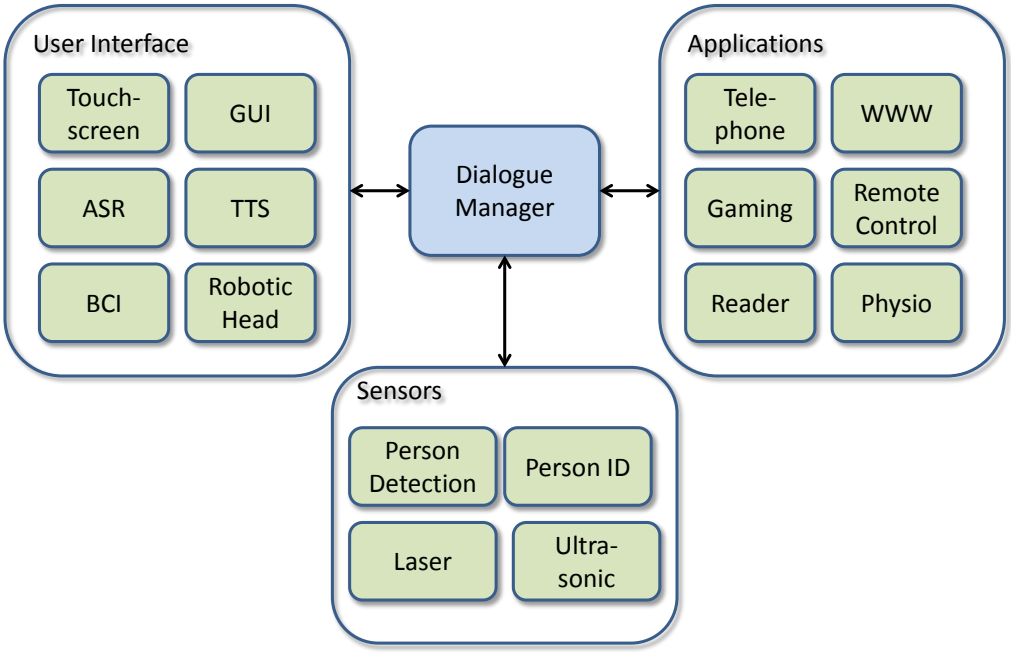


Figure 2.2: Overview over all software modules connected to the dialogue manager.

towards the user. Another possibility to control the robot is the Brain-Computer-Interface (BCI). The BCI uses relations between visual stimuli and measured brain response.

Several sensor units are connected to the dialogue manager. Person detection and recognition helps the dialogue system to locate and identify the user. Persons are detected using laser-based leg pair detection combined with face detection in camera images. The face detection module is also used for face identification. Additionally, speaker recognition may also be used for person identification. During navigation, laser and ultrasonic sensors are used for localisation of the robot within its environment.

Various applications are integrated, whereby all of them can be controlled by the dialogue manager. Several modules for communication have been developed. A telephone function capable of performing video calls is available. Using an alarm call function, a video call connection to a caretaker can be established. A web browser is integrated into the GUI. In particular, a web application enabling either to discover upcoming events happening nearby or to relive past attended events is installed. For entertainment purposes, several games can be played on the touch-screen. Furthermore, the loudspeakers can be used to play audiobooks. Similar to the audiobook player, a reader functionality can be used to read text from a real document that is being held in front of the camera. A module for physiological monitoring is available for recording and storing of health data.

3 Hardware Components

This chapter gives an overview over the hardware components which are involved in the dialogue system. The robotic platform is based on a SCITOS G5 monile robot base platform from MetraLabs. As introducted in Chapter 2, the robot can mainly be divided into a driving unit and an interaction unit. The driving unit uses a differential drive, which enables the robot to drive up to $1.0\ m/s$. Within the driving unit, an industrial embedded PC is mounted which controls the hardware components of the robot. Additionally, a PC running Windows controls the dialogue manager and the touch screen. The touch screen serves two purposes. On the one hand, its touch functionality is used as an input modality to control the robot. On the other hand, the graphical user interface (GUI) is displayed on this screen. The non-glare display has a diagonal of 15". Touch functionality is implemented with resistive technology. Around the touch-screen, in the corners of the case, four microphones are mounted. These microphones are connected to the Windows PC and are used to record environmental sounds and speech. Thereby, the audio signal is used for speaker recognition and for speech recognition, which are both crucial parts of the dialogue system. Additional hardware components that are mounted on the robot include a Wii gaming console, a TV card, a Kinect sensor, and some ground lights.

Above the touch screen display, the robotic head is mounted. Figure 3.1 shows a schematic drawing of the robotic head. The head has several degrees of freedom. It can be turned $360°$ horizontally and vertically up $(15°)$ and down $(6°)$. The two synchronized movable eyes can be turned up to $8.5°$ in both horizontal directions and the eye lids can be opened or closed (independent of each other) on a continuous scale. Above the eyes, a row of blue LEDs is mounted, which can be turn on and off or set running or blinking with any frequency. Additionally, the brightness of the LEDs can be controlled. In total, this sums up to 6 degrees of freedom which are used to display different facial expressions. On top

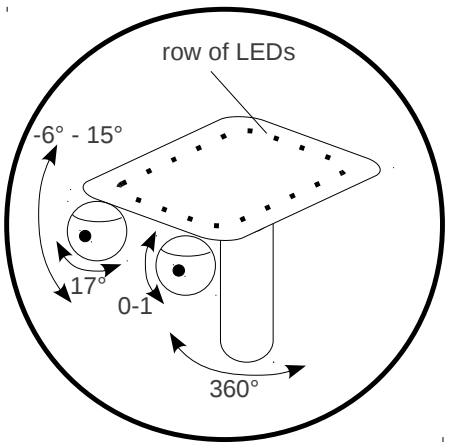


Figure 3.1: Schematic drawing of the robotic head

of the head, an omnidirectional camera is mounted, which delivers a 360° image. Due to its mounting position, the main purpose of this camera is to localize and identify persons using face detection.

Over the course of the project, some modification of the platform have been done. A new microphone has been placed on the robot's neck. This helps to increase audio recording quality, because the microphones around the touch screen are very sensitive to all kinds of interference. Furthermore, for example, a kinect sensor was integrated, which is used for person detection.

4 Dialogue System Components

In this chapter, all software modules which are connected with the dialogue system are described.

4.1 Dialogue Manager

The dialogue manager (DM) is a central module of the ALIAS system which controls the final behaviour of the robot. The DM receives user inputs and other events from the various other modules, processes these inputs and events, decides about appropriate actions, and sends the corresponding outputs or commands to the respective modules. The dialogue manager prototype has been described in deliverable D3.4 [2]. To summarize, functionally, the DM implements a mapping from events (e.g. "Battery empty") and user inputs (e.g. from the speech recognizer) to result signals (to control the robot, or to give visual and/or natural language output to the user). In order to produce a consistent behavior when responding to inputs from the rest of the system, event processing is serialized. For each input received, the DM decides on a suitable response behavior and activates it by sending a command to the respective module. The dialogue manager has been realized in two parts: the DM Core operates on a uniform conceptual representation that maps the input events to some suitable output actions, whereas the DM Communicator deals with the communication protocols and translates from and to the external data formats. Technically, these two parts are realized as separate applications and processes, communicating via an XML-RPC protocol. The DM Communicator manages the communication between the DM Core and the other components of the robot. To communicate with the other components, various protocols and data formats are used (see deliverable D3.7 [5]). The Communication-Engine of the DM Communicator uses different adapters to abstract from this diversity by translating different input signals into a uniform conceptual event representation and by translating DM Core action decisions into different outgoing signal formats (see Figure 7). The DM Core encompasses two major components, the NLP-Engine and the Decision Engine. The natural language processing engine (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 DM Communicator and decides on the robot behavior by choosing the next action. To choose the appropriate action, the Decision-Engine takes into account the current situation, which is based on the history of past events.

This section describes the Dialogue Manager architecture and implementation (section 4.1.1) and the interfaces to the other robot components (section 4.1.2).

4.1.1 Architecture

The Dialogue Manager is the central communication and decision component of the robot. The Dialogue Manager has two main parts: the Dialogue Manager Communicator (DM Communicator) and the Dialogue Manager Core (DM Core). Figure 4.1 shows the architecture.

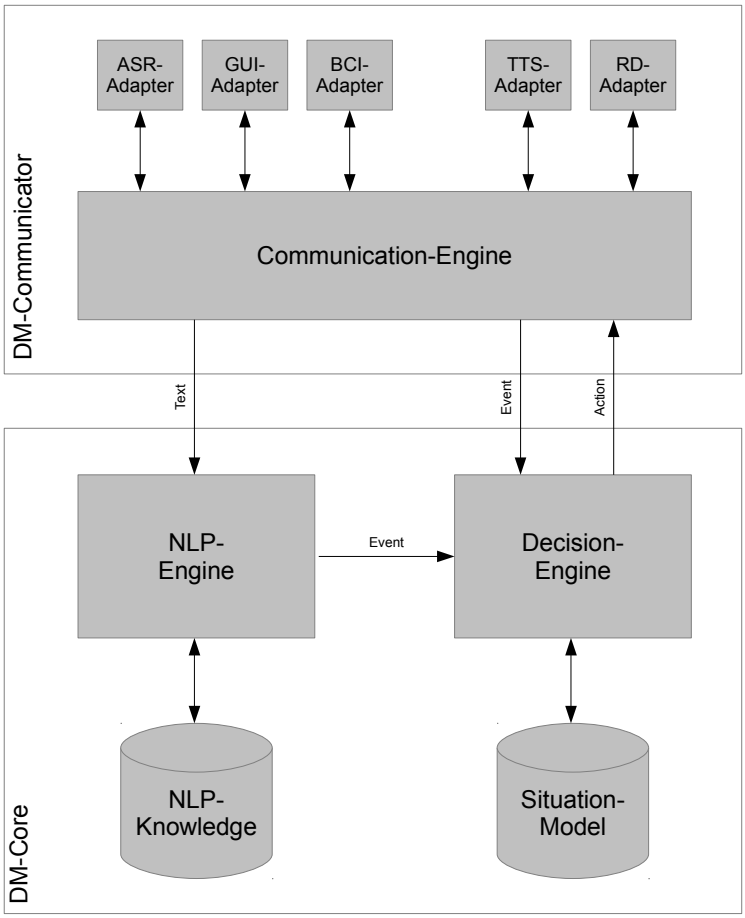


Figure 4.1: Dialogue Manager architecture

The DM Communicator manages the communication between the DM Core and the other components of the robot. To communicate with the other components, various protocols and data formats are used (see section 4.1.2). 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 NLP-Engine and the Decision Engine. The natural language processing engine (NLP-Engine) translates spoken user in-

put 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 DM Communicator and decides on the robot behaviour by choosing the next action. To choose the appropriate action, the Decision-Engine takes into account the current situation, which is based on the history of past events.

To fulfil its tasks, the DM Core uses two knowledge databases. The NLP knowledge base provides general information regarding the world and expert knowledge about the life of elderly people and to the translation of natural language to abstract event entities. The situation model knowledge base stores information needed by the Decision-Engine to compute the new situation and to choose the next action, both 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.

4.1.2 Interfaces

The interfaces to the other robot components are implemented as communication adapters that are part of DM Communicator. This section lists all currently 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 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 4.1 shows an example JSON array.

Listing 4.1: "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 syllables are represented by `00V`.

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. The default language of the TTS system is English. 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 4.1 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 to send a corresponding command packet to the GUI. After starting the web browser, the GUI acknowledges the command with a signal packet.

Commands are send from DM Communicator to the ALIAS GUI to initiate actions, e.g. starting a game or changing the current 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 4.2 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, 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 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 main menu. In turn, the GUI acknowledges each command packet with a corresponding signal packet.

Table 4.1: 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

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 of 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 4.3 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 a `EVENT` packet to the DM Communicator which indicates whether the robot could reach its target or not.

Table 4.4 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 in idle state. when the user orders the robot to approach him via speech input, 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, step 5 indicates that the robot has reached its target.

Table 4.2: 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"/>`
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 4.3: 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

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 [4]. The DM Communicator acts as such a controller by sending the BCI XML documents that define 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 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. Currently, several masks are implemented: A main mask, and masks to control skype and audiobooks.

Listing 4.2 shows the XML document describing the BCI main mask. This mask is ini-

Table 4.4: 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"/>

tially transmitted from DM Communicator to the BCI component during the handshake procedure.

Listing 4.2: 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>
      <Port>22346</Port>
    </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">
          <Instruction>run</Instruction>
          <ICONPosition>[1,1]</ICONPosition>
          <DispSymbol>
            <Text>Skype</Text>
          </DispSymbol>
          <CommType>multi</CommType>
          <CommandParameter>
            <Parameter type="application">skype</Parameter>
          </CommandParameter>
        </SingleCommand>
        <SingleCommand CommandName="StartAudioBooks">
          <Instruction>run</Instruction>
          <ICONPosition>[2,1]</ICONPosition>
          <DispSymbol>
            <Text>Books</Text>
          </DispSymbol>
          <CommType>multi</CommType>
          <CommandParameter>
            <Parameter type="application">audiobooks</Parameter>
```

```
        </CommandParameter>
      </SingleCommand>
      <SingleCommand CommandName="BrowseTheWeb">
        <Instruction>run</Instruction>
        <ICONPosition>[2,3]</ICONPosition>
        <DispSymbol>
          <Text>www</Text>
        </DispSymbol>
        <CommType>multi</CommType>
        <CommandParameter>
          <Parameter type="application">webbrowser</Parameter>
        </CommandParameter>
      </SingleCommand>
  </Application>
  </ControlGroup>
</AppList>
```

Listing 4.3 shows a message send from BCI to DM Communicator during a skype chat. The user entered the word HALLO and send the message.

Listing 4.3: 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 [4].

4.2 User Interface

Several modules are used for communicating with the user. All communication channels are controlled by the Dialogue Manager.

4.2.1 Input Modules

There are three primary input channels: Speech recognition, touch screen display and brain computer interface (BCI).

A module for large-vocabulary speech recognition ensures easy handling of the robot. The grammar of the system for automatic speech recognition (ASR) includes commands to navigate through the GUI and to start applications. Furthermore, the robot can be controlled by speech commands, e. g. starting a navigation task.

The touch screen display can directly be used to navigate through the menu containing the applications. All applications available in the menu can directly be started by clicking on them. This includes entertainment applications like gaming or web browser. Security functions like starting an alarm call to a doctor can also by triggered through the touch screen.

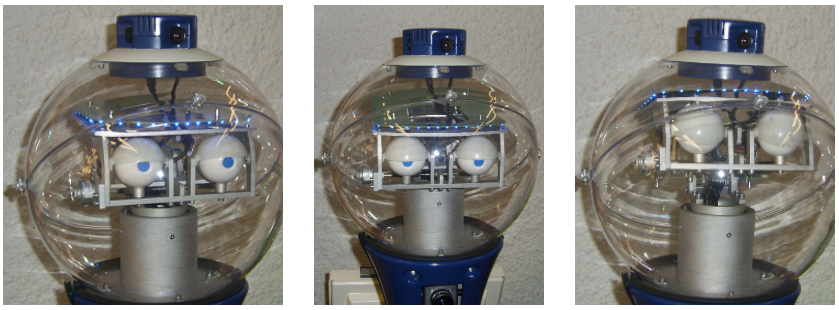
With the BCI, it is also possible to navigate through the GUI and start modules. It ensures the possibility to handle the robot for users like stroke patients.

4.2.2 Output Modules

Several modules are used to give feedback to the user. The primary output module is the GUI. The current status of the menu is displayed on the touch display. Furthermore, several applications are directly linked to the GUI. This includes most of the entertainment applications like games, or audio books.

In order to give replies to speech commands, a speech synthesis module is integrated. When ASR is used to navigate through the menu, the robot answers with speech commands to communicate its new status to the user.

As an additional module for generating feedback to the user, the robotic head is used to display different facial expressions. Figure 4.2 shows three exemplary facial expressions.



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

Figure 4.2: Different facial expressions of the robotic head of the ALIAS platform.

4.2.3 Sensor Modules

Several of the sensor modules of the robot deliver their input to the dialogue manager.

Person Detection

The person detection module combines face detection with laser-based leg-pair detection. Both face detection and leg-pair detection are running in parallel and their output is combined to create a more stable and robust person detection hypothesis.

The face detection system has been implemented for the face identification system of the ALIAS platform [6]. It uses the omni-directional camera which is mounted on top of the robotic head. This camera provides a 360-degree image with a resolution of 720×280 pixels which enables the detection of faces in all directions of the robot. To detect faces,

the algorithm introduced by Viola and Jones [15] is used. This algorithm is characterized by its very effective and fast processing and achievement of 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) [3] 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. Afterwards, 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. When multiple users are around the robot, mostly the person nearest to the camera will be detected.

The second person detection channel is the laser range finder on the robot. It is able to detect legs, if they are within the laser scan, and works fine up to 5-8 meters (which is sufficient for home environments), but also detects a lot of leg-hypotheses, which are actually no legs. To improve the reliability and robustness of the face-detection approach and the leg-detection approach, both hypotheses channels are fused.

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-relative metric information, where a person is in terms of angle and distance of the person, both channels could 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, which are very close to each other, are given 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-alarm 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.

Speaker Identification

The Speaker identification module implemented on the ALIAS Robot is based on ALIZE/LIA_RAL [1] system using SPRO [7] for front end feature extraction. The block diagram of the speaker identification module is shown in figure 4.3. The system is based on the standard Gaussian mixture model(GMM) using a universal background model (UBM) paradigm [14].

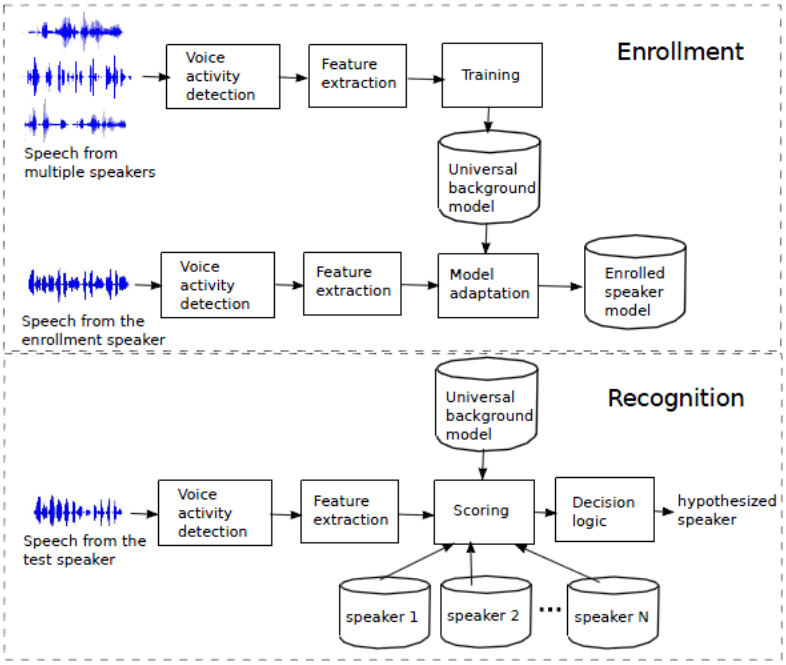


Figure 4.3: Block diagram of speaker identification module

Face Identification

The face identification module of ALIAS uses the Viola-Jones Algorithm [15] for face detection and eigenfaces for face identification. For face detection, the same module is used as for person detection. The system fetches an image from the camera and first of all transforms the image to greyscale, since the Eigenface approach is done on greyscale images. Afterwards, the face detection is started. This is done with the Haar cascade classifier (Viola- Jones Algorithm). Only the hypothesis with the highest probability is returned, in the form of a rectangle as a bounding box of the face. The image segment containing the face is then normalized in contrast and brightness, to minimize the influence of the lighting conditions. Then, the face identification is started. For this purpose, the image segment is projected onto the PCA subspace. The identity is then determined by classifying the unknown face with a nearest neighbor classifier.

4.3 Applications

Various different applications have been integrated into the framework of ALIAS. Most of them can be categorized as entertainment or communication applications.

4.3.1 Reader Functionality

In deliverable D1.2 [11] a list of functionalities desired by the users was presented. On this list is the functionality that supports reading of digital books and texts obtained from

a webcam. In Figure 4.4a the main menu of the system is shown, it can be selected between the reading of digital books (see Figure 4.4b) and the webcam text scanning (see Figure 4.4c). The reading of digital books supports the epub-format, these books can be loaded by system and then are presented with in the window. It is possible to address specific chapters or lines of the book. In addition it is possible to adjust the font size of the text. For the reading of the text different voices (female, male) can be chosen. The buttons of the window were optimized for touchscreen use. The text scanning function shows a window with the current image of the webcam. Depending on the selected voice (German, English and French are available) different language vocabularies are used to support the OCR. The scanned text is shown in a window on the right side. Due to the fact that the OCR has sometimes difficulties with recognizing letters (e.g. an 1 is sometimes interpreted as l) it is possible to adjust the text before starting the reading function.

4.3.2 Games

The graphical user interface contains a small selection of games that is accessible via the games menu, see Figure 4.5. A few common games have been selected for integration. This includes Chess, Sudoku, Solitaire, and Tic-Tac-Toe. The games can be started either by touching the corresponding icon on the touch screen or by using a speech command. Additionally, games can also be started by the BCI.

4.3.3 Audio Books

Another functionality which can be started from the menu is playback of audio books. Using the GUI, speech commands or BCI, it is possible to display the list of available audio books. From this list, the desired audio book can be chosen and started. The contents of the audio book is then played over the loudspeakers of the robotic platform.

4.3.4 TV

Using the TV card mounted on the robot, it is possible to watch TV within the GUI. The signal from the TV card is forwarded to the PC connected to the GUI. The TV function can be started from the menu.

4.3.5 Internet Browser

For communication and entertainment purposes, a web browser is integrated in the GUI. It provides a platform for the web-based services, like the Event Search Engine (ALIAS WP4). There have been several attempts using different techniques to achieve proper web-browser integration. So far, integrating Firefox or Microsoft Internet Explorer has been tested. Furthermore, developing a browser fully integrated within the GUI, using QtWebKit has been considered. The current version of the GUI uses the QtWebKit-based browser.

4.3.6 EventMedia

Events are a natural way for referring to any observable occurrence grouping persons, places, times and activities. They are also observable experiences that are often documented by people through different media (e.g. videos and photos). This intrinsic connection between media and experiences is explored in two aspects with the EventMedia application. Firstly, a method for finding media hosted on Flickr that can be associated to a public event is applied. It uses linked data technologies for enriching semantically the descriptions of both events and media, so that people can search and browse through content using a familiar event perspective. Secondly, another method is used to automatically detect and identify events from social media sharing web sites. The approach is based on the observation that many photos and videos are taken and shared when events occur. The approach focusses on detecting events from the spatial and temporal labeled social media, and an algorithm is used to retrieve those media, perform event detection and identification, and finally enrich the detected events with visual summaries. The EventMedia application is described in detail in Deliverable D4.2 [9].

4.3.7 Telephone

Based on Skype, a video telephony functionality is integrated into the GUI. In order to achieve the highest user acceptance it is advisable to incorporate an already well-established telephone system instead of developing a new ALIAS-specific solution. In this area the Skype communication system seems the most viable option. And in addition, Skype offers APIs for integration to external applications like the ALIAS GUI. Basically there are two different Skype APIs, the Desktop API and the SkypeKit API, supporting varying levels and modes of integration. A module providing a telephone functionality is fully integrated into the GUI. It can be started by navigating through the GUI or with speech commands.

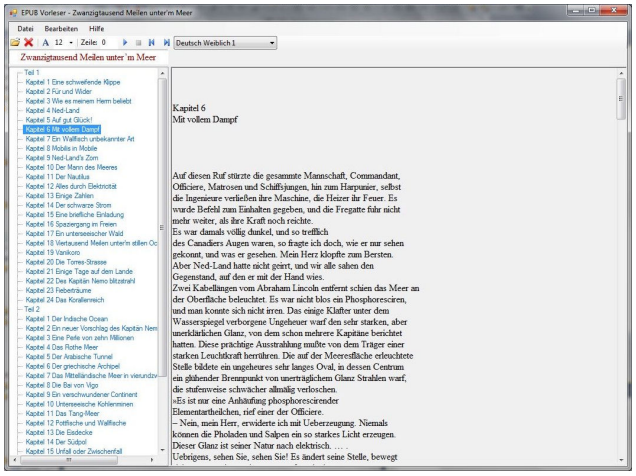
4.3.8 Physiological Monitoring

For the ALIAS-system a physiological monitoring module was developed. This module can support live-data obtained from two different sensor devices and the manual entering of data. In addition to the four vital signs (heart rate, blood pressure, respiration and temperature) the weight and blood sugar can be stored in the database. The physiological monitoring module is designed as a web-application therefore the interaction with the system takes place with the webbrowser. The reason behind this design approach is that authorized persons (e.g. doctors, relatives) can have access to these data to check the current health status of the ALIAS user. The system supports multiple users and has two different styles of data presentation (table and graph). More details about the system can be found in D3.5 [12] and D3.6 [13]. As a consequence of the midterm review it was decided, because of data security, to omit the physiological monitoring module in the final dialog system, therefore no further activities in this direction were taken.

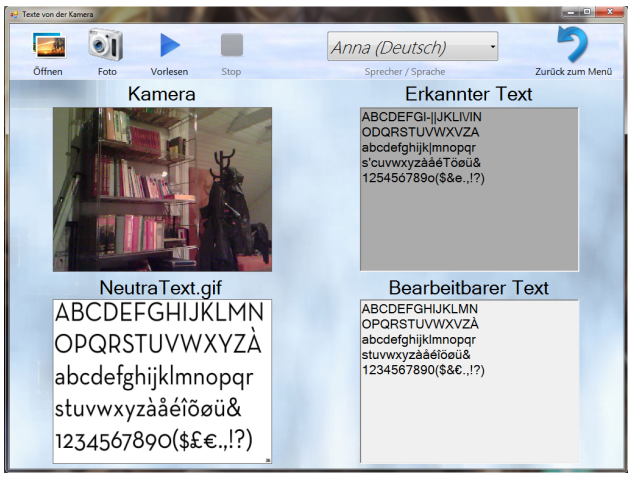
4.3.9 Remote Control

In combination with the video telephony functionality, the possibility to control the robot remotely has been developed for ALIAS. In the Alarm Call use-case scenario, the user can establish a video telephony connection to a relative or carer in an emergency situation. The carer can then remotely control ALIAS with his keyboard. Thereby, the remote person can inspect the situation and navigate to the user remotely, using the provided camera image.

(a)



(b)



(c)

Figure 4.4: Reader Functionality Overview.

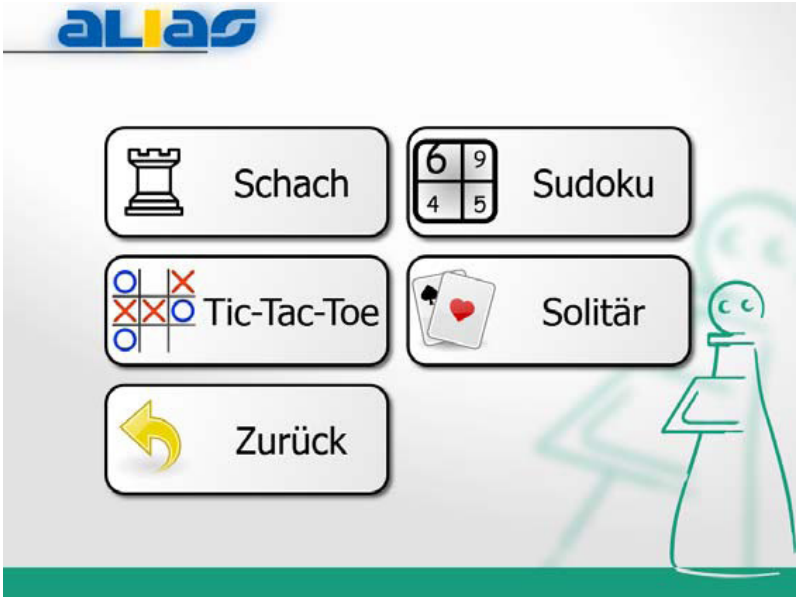


Figure 4.5: The games menu of the ALIAS GUI

5 Use-case scenarios

Based on the functions that were implemented for the robot, several use-case scenario demonstrations have been prepared. These use-cases demonstrate the functionality of the implemented modules. In all of these use-cases, the dialogue systems plays an important role. A detailed description of the use-cases is given in Deliverable D7.3 [10].

5.1 Self-experience scenario

The self-experience scenario was developed for testing the GUI and its integrated functionalities. In this scenario, the user should try to use the GUI, without too much instructions. One out of three simple high-level instructions is given to the user:

- Find the list of games, start "‘tic-tac-toe"’ and play a short match.
- Find the telephone functionality and start a telephone call.
- Find and start the Internet browser.

The user is allowed to use the touch screen as well as voice commands. It is then observed, how the user approaches his goal and if he succeeds to reach this goal.

5.2 Remote Control by Secondary User

Elderly people can request assistance by relatives or caregivers using a Skype call and asking the called person to remote control the robot. There are many possible situations where the user might ask somebody to remote control the robot to visually check something. For example, there might be some unusual noise in the kitchen and the elderly is not able to locate the reason of the noise. The elderly asks ALIAS to call a relative via Skype. During the Skype call, the elderly reports the unusual noise and asks the relative to use the remote control to move ALIAS into the kitchen and to have a look for the reason causing the noise. In another scenario, the elderly might ask a caregiver to check a skin eruption using the remote control.

5.3 BCI Scenario

The basic idea of this scenario is to test the BCI system, which replaces speech input and touch screen by BCI mask. In general, the self-experience s cenario is repeated, using the BCI to control the robot instead of the touch screen or speech commands.

5.4 Ground Lighting Scenario

The main idea of this scenario is the guiding of a person at night, when no light is present. One example is that the robot could guide a user at night to the toilet or to the kitchen. The robot can light up the space behind itself, so the person can observe the ground when following the robot. To have some control about the guiding process, the robot can be controlled by speech commands like "slow down, drive faster, stop" or "abort operation".

5.5 Event scenario

Finding pictures or videos that we once shared on the Web is already a challenge. So giving them a second life seems unthinkable. But this is one of the objectives of the EventMedia application installed on ALIAS.

People are sharing an increasing amount of multimedia content on the Web. But these millions of photos and videos that we put each year on sharing sites or social networks often remain unused. At best, they are classified and marked, but further indexing is difficult. Recognition tools which could help find faces and places automatically are limited. The term "media" is to be understood in the broad sense. It can refer to videos and photos as well as to tweets, Facebook or audio messages. All this multimedia content corresponds to events and we found it more efficient to find them through an event search, by place, date or event category such as a concert, an exhibit, a soccer game, a birthday or a party with friends. In this scenario, the elderly user should use the EventMedia application on the ALIAS robot to find information about past and upcoming events.

6 Conclusions

In this deliverable, the final dialogue system of the robotic assistant ALIAS was described. Built around the dialogue manager, all user input and output is processed by this system component. Input from the ASR module is interpreted and corresponding actions are triggered, e. g. to start another module. The dialogue system is the core module of five described use cases. These use cases have been tested in field trials with elderly users.

Bibliography

[1] 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.

[2] A. Franke. D3.4 - Documented prototype of dialogue manager (software). ALIAS Deliverable D3.4, 2011.

[3] 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.

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

[5] J. Geiger, M. Pasdziernik, R. Vipperla, and J. Kessler. D3.7 - Dialogue System Updated to the User's Needs. ALIAS Deliverable D3.7, 2012.

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

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

[8] J. Kessler, A. Scheidig, and H.-M. Gross. Approaching a person in a socially acceptable manner using expanding random trees. In *Proceedings 5th European Conference on Mobile Robots*, pages 95–100, 2011.

[9] X. Liu, R. Troncy, and B. Huet. D4.2 - Module for cross-media linking of personal events to web content, v1. ALIAS Deliverable D4.2, 2011.

[10] C. Martin. D7.3 - Second ALIAS prototype. ALIAS Deliverable D3.3, 2012.

[11] S. Niedermeier, K. Scheibl, W. Schneider, S. Ihsen, F. Kohl, P. Dinckelacker, S. Glende, and R. Troncy. D1.2 - List of selected functions. ALIAS Deliverable D1.2, 2011.

[12] T. Rehrl and J. Geiger. D3.5 - Documented physiological monitoring system. ALIAS Deliverable D3.5, 2011.

[13] T. Rehrl, J. Geiger, M. Golcar, S. Gentsch, and J. Knobloch. D3.6 - First dialogue system with integrated physiological monitoring. ALIAS Deliverable D3.6, 2011.

[14] 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.

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