Project FoSIBLE

Fostering Social Interactions for a Better Life of the Elderly

Deliverable

D5.1: Report on requirements of software components and tools for observation and tracking in laboratory and Living Lab environment. Delivery date: M16

Responsible

AIT (Lead Contractor)

Participants

AIT, CURE

Abstract

D5.1 contains the requirements of software components and tools for observation and tracking in laboratory and Living Lab environmentof the FoSIBLE project. The overall

FoSIBLEsystem as well as the software specifications for the components for Observation and Tracking, Gesture Recognition and Novel Input Devices are described and discussed.

Table of Content

1 Introduction

1.1 Purpose of the Document

The purpose of this document is to define the requirements of software components and tools for observation and tracking in laboratory and Living Lab environmentof the FoSIBLE project. The software specificationsfor the components for Observation and Tracking, Gesture Recognition and Novel Input Devices are described and discussed.

1.2 Definitions, Acronyms and Abbreviations

Acronym	Description
FoSIBLE	Fostering Social Interaction for the Well-Being of the Elderly
IF	Interface
PC	Personal Computer
AE(R)	Address Event (Representation)
3D Sensor	Devices that delivers spatial (three dimensional 3D) information of a given scene , typical x,y and depth-z
SW	Software
HW	Hardware
OS	Operating System

2 System description

2.1 General system description

The FoSIBLE System allows the interaction of elderly people via a social media platform. A central server will host a databroker, database and webserver which comprise the media platform. Several components are installed in the end-user's homes which run user-interfaces to allow the interaction with the FoSIBLE social platform.

Commands can be sent to the Smart-TV (HBBTV) using gestures or a Tablet PC. The user has the free choice to perform simple gestures for simple interactions with the TV (e.g. change channel, etc.) or use the Tablet PC for more severe actions.

The gesture Recognition Components are a Mediacenter PC running Microsoft Windows and the two 3D sensors UCOS and Microsoft Kinect. The two sensors use a different technological approach to generate 3D depth data. The parallel use of these technologies is primarily for evaluation.

The PC is used to run the Software, which will process the sensor data to recognize the gestures (Gesture Recognition Module). Probability values are handed over to the Gesture Interpreter which will decide the meaning of the gesture and send a command to the correct receiver (TV or databroker).

The databroker is the central data handling component of the project. Data from all input devices (Tablet-PC, gesture control) and other sensors are collected in a central database. Part of this data are used as content for the Social media platform, which itself can be viewed either on the Smart-TV or the Tablet-PC.

A network connection (LAN/WLAN) is the primary communication media between the hardware components of the system.

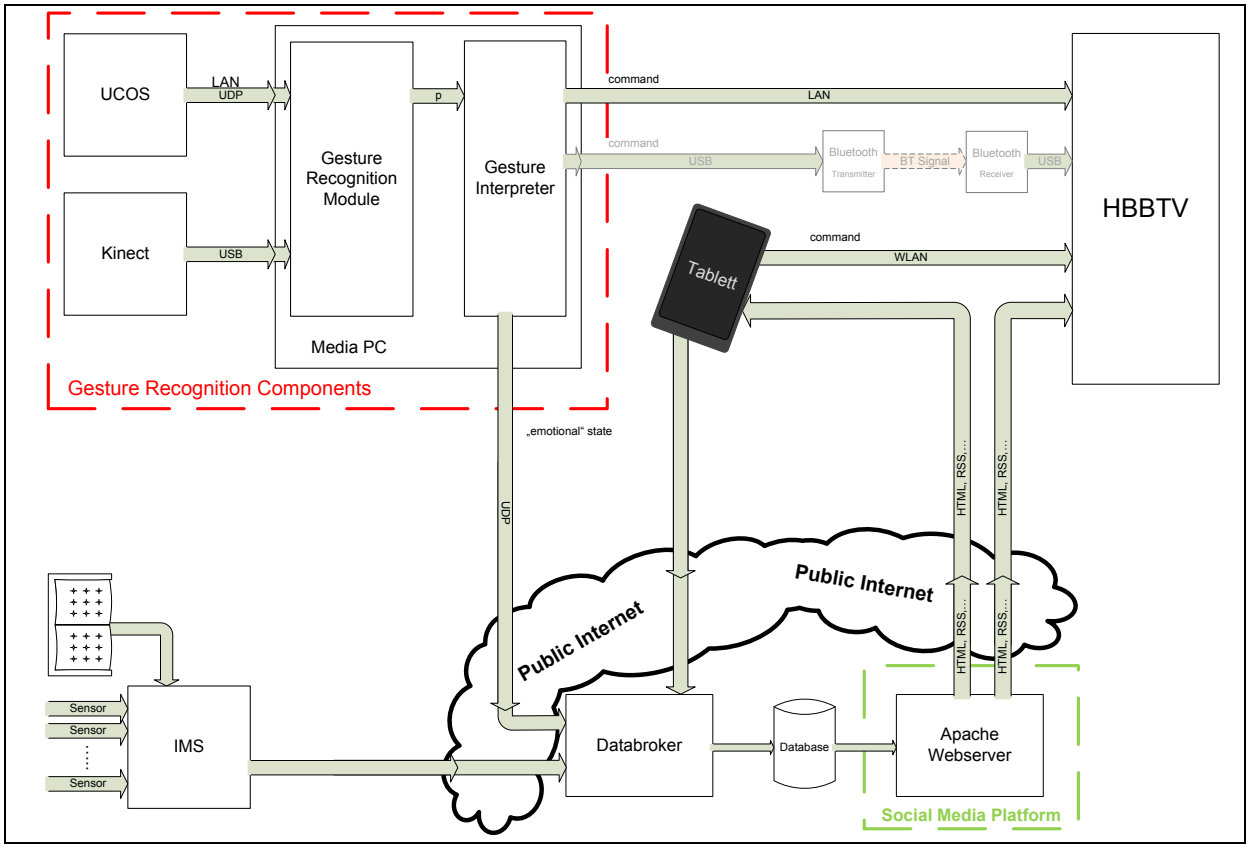

Figure 1: OverallFoSIBLE block diagram

2.1.1 Observation and Tracking

The number of people in the room where the social media TV is located will be sensed by a 3D device. From this information a "social and emotional state" of the end user will be estimated. This information will be forwarded to the social media platform server via a databroker.This will allow to estimate if a user is often alone or with company and from this other users can be informed and motivated to join that person (more often) via the social platform. For the observation and tracking function only the UCOS sensor or motion sensors are foreseen.

2.1.2 Gesture Recognition

The Gesture Recognition Modules are used to generate commands to control the menu of application and/or widgets of the social media platform. If feasible gesture commands can also trigger Content updates. Two different 3D Sensors are evaluated for the gesture recognition, Kinect and UCOS. These sensors are connected to a PC which will run the gesture recognition software and distribute the commands between the receivers (TV set or databroker).

The intention of integrating gestures for controlling the FoSIBLE application is to offer an alternative to the tablet used as main input device. The users don't need to grab anything but can immediately start to interact with the system by just moving their hands. This is why

a simple trigger for initiating the gesture control is necessary e.g. waving a hand looking directly at the TV.

Gestures are referred to as natural input modality that evolved from real-world interaction stylesbeing more intuitive and easier to learn than indirect input modalities like remote controls. However, gestures need to be based on a good world-model to provide usability benefits [3]. Otherwise the interaction becomes indirect again and due to themissing affordance (if the users don't get any hints what options are available) the users have to learn and remember non-intuitive gestures. When complex gestures are used intensively another drawback reported is fatigue– especially when the hands need to be hold too long in the air. Due to declining physical and cognitive abilities these drawbacks affect older people in particular.

For this reason not all kind of gestures are feasible for the use in FoSIBLE. A classification of gesture based human computer interactions was presented by Karam et al.[4]. Figure 2 presents this classification that not only involves gesture styles but also application domains, enabling technologies and system response. However, the elements of those last mentioned categories don't belong into the scope of this document and were already discussed in the work plan orprevious deliverables respectively.

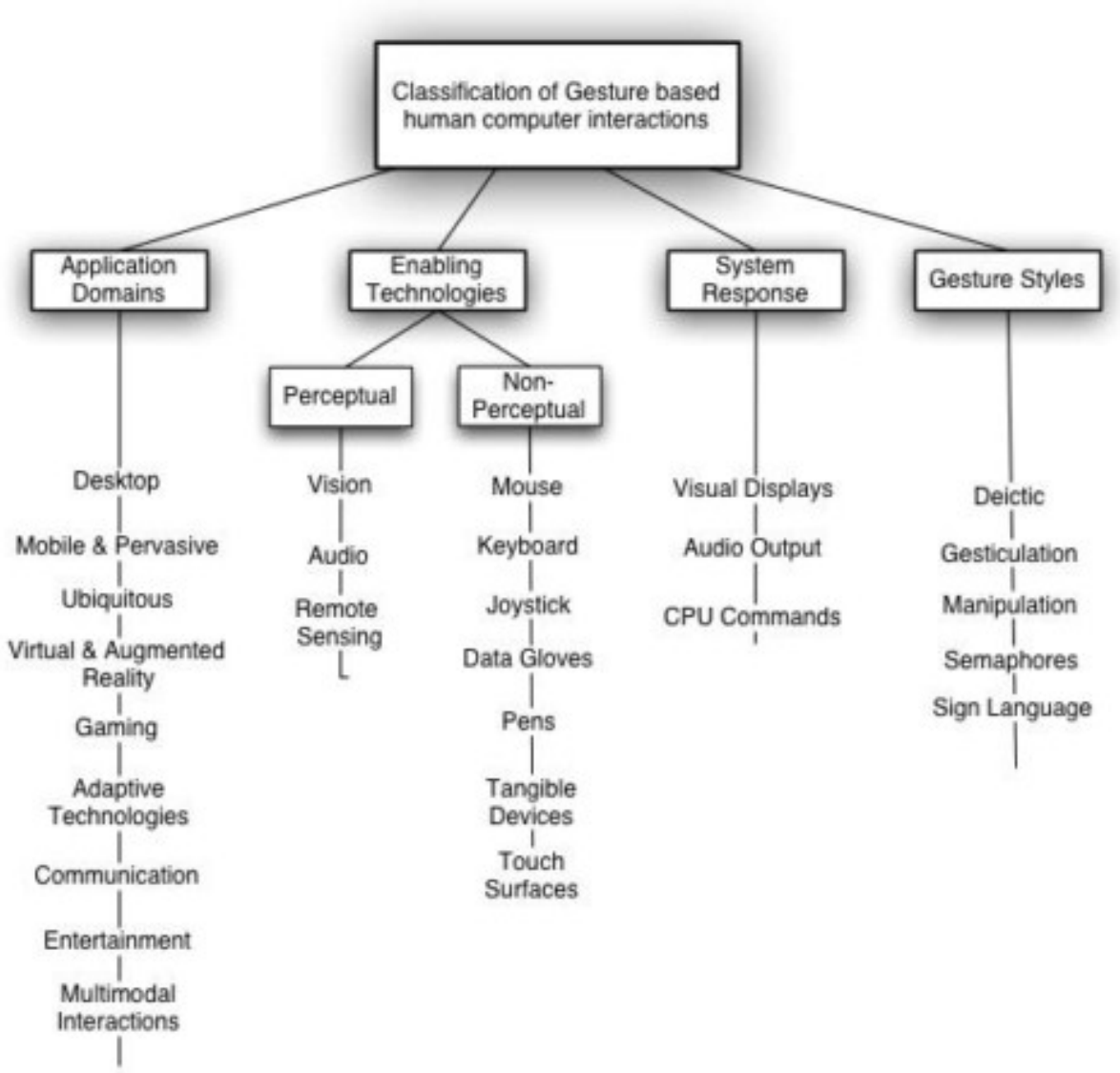

Figure 2: This diagram visualises the taxonomy of gesturebasedhuman computer interactions. It is based on a research review and has beencreated by Karam et. al[4].

According to [4] gesture styles for interaction can be classified into five categories: gesticulation, manipulations, semaphores, deictic and language gestures. While Gesticulation and language gestures don't fit into the context of FoSIBLE semaphores, manipulations and deictic gestures could be applicable in theory.

Semaphoric gestures comprise any gesturing system that employs a stylized dictionary of static or dynamic hand or arm gestures [4]. In the context of FoSIBLEeach gesture out of this dictionary could trigger a specific function of the social media platform. Examples of those gestures are presented in Figure 3. However, as they are not natural users must learn the defined set of gestures leading to an additional barrier not only for older people [6]. Even if there are just few semaphoric gestures learning and recalling those gestures can be anexhausting task due to the lack of affordance.

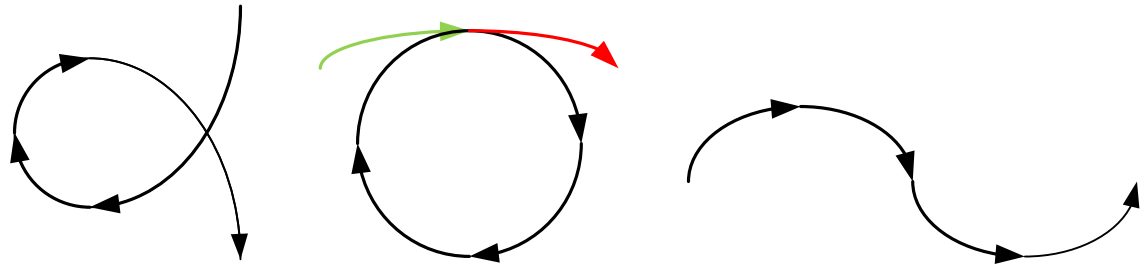

Figure 3:Examples of semaphoric“short cut” gestures (not selected for use in the project)

Deictic gestures involve pointing to identify the spatial location of an object within the context of the application domain [4]. In the context of FoSIBLE this kind of gestures can be used for e.g. menu interactions or other selection tasks. Deictic gestures have been used in many human computer interactions in various application domains. However, pointing at a certain menu item could be difficult for older people e.g. if the target item is too small or if theydon’t have the physical abilities anymore to hold the arm stable.

The purpose of manipulative gestures is to control an entity by applying a tight relationship between the actual movements of the gesturing hand/arm with the entity being manipulated [4]. The main output device in FoSIBLE is a TV so gesturing, either taking place in two dimensional or in three dimensional space, applies to a flat target device. According to the FoSIBLE scenarios described in D2.2 there is no need of three dimensional inputs and due to the higher complexity of three dimensional gestures we focus on gesturing plain in front of the TV. The quality of this kind of gesturing depends like the usage of deictic gestures enormously from the design of the user interface.

In order to provide easy access to the FoSIBLE system without the need of remote control menu navigation is an important task for the gesture control. By controlling the system via gestures the user gets a quick and easy-to-use access to news and information of his community. Some ideas of the accordant menu user interface are presented in **Fehler! Verweisquelle konnte nicht gefunden werden.**. For more complex tasks (e.g. entering text) gestures are not appropriate. For being a suitable alternative to the menu navigation via tablet, the gesture control of the FoSIBLE application needs to perform: (i) recognition of the initiation of a new interaction, (ii) targeting at certain user interface elements and (iii) triggering of selections.

Researches and developers invented various different kinds for menu navigation via gestures. Some of them are presented and evaluated by Chertoff et al. [1] and Karam et al. [4]. However, research on the performance of elderly people with non-touch freehand gesture based interactions like intended for FoSIBLE hardly exists. For this reason we conduct some user tests in order to gain knowledge of which kind of menu and which kind of gesturing is (i) feasible and (ii) preferred by older people. Details about these tests with deictic and manipulative gestures are described in section 4.3.

Considering the work done for gesture based interactions in general one can state that there is a lack of collaboration between technicians who are focussed on improving gesture tracking algorithms and user interface designers eager to define and evaluate new forms of gestures. This leads often to either new algorithms not yet extensively evaluated with users or interaction prototypes not including the latest technologies. In the scope of the FoSIBLE project AIT and CURE are working closely together for combining both aspects seamlessly.

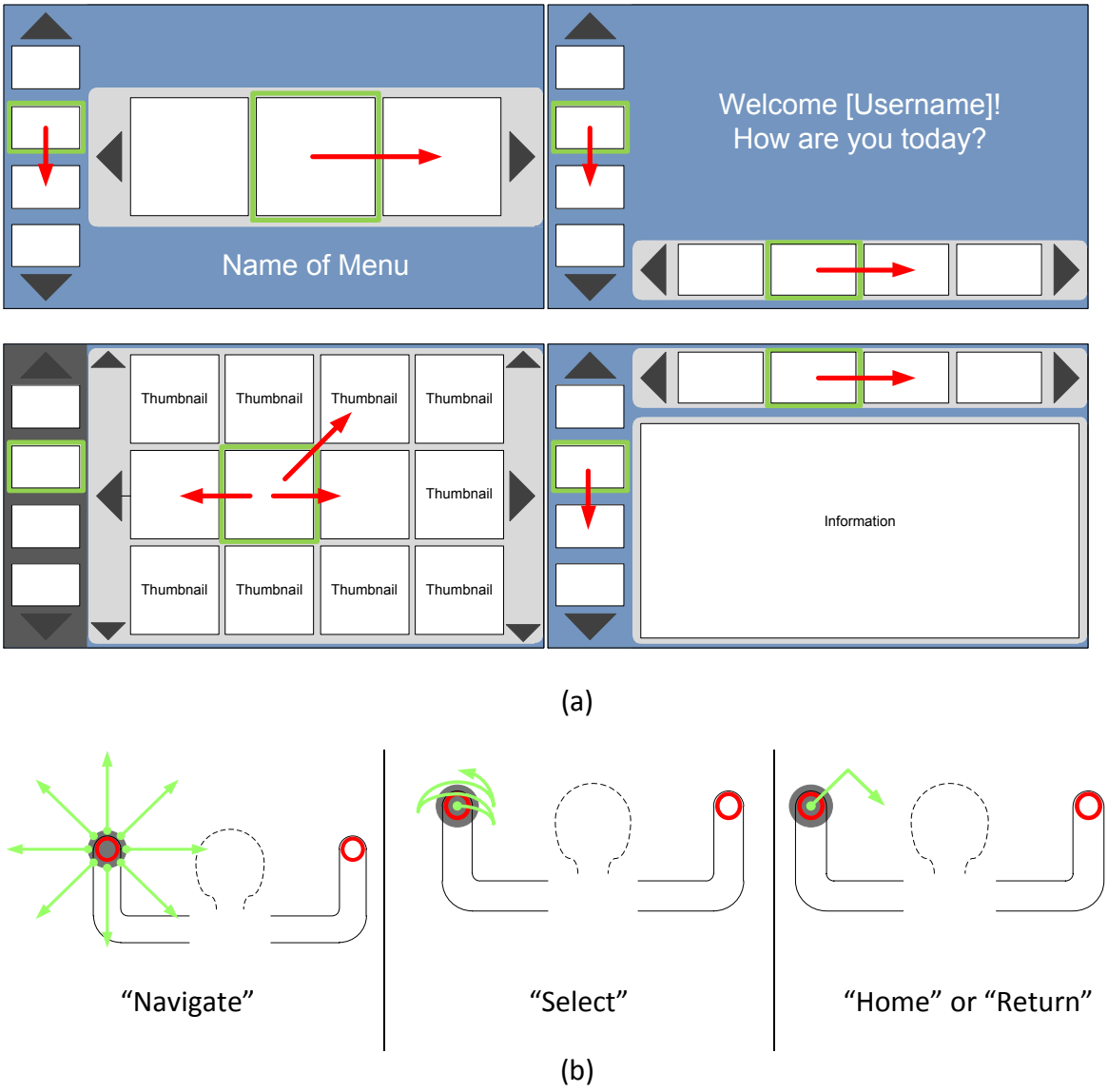

Figure 4: (a) Examples of typical simple menu designs und user interfaces for gesture controlled menu navigation (selected for use in the project) (b) Corresponding selected Deictic gestures for menu navigation.

2.1.3 User-system interaction through novel input devices

Although gestures can be used for most navigation and selection tasks, not all inputs can be done using gestures alone. Furthermore, some users might not want to use gestures at all. Thus, it was decided to provide an additional input terminal to the users, which they can use

to navigate on the TV, but also to enter text (e.g. during chats or when writing a short article). As traditional input devices like keyboards and mice are quite cumbersome to use in a living-room environment, it was decided to use a tablet PC. The tablet can additionally be used to display messages, when the TV is switched of.
The tablet is connected to the data-broker, the web-server, and the HBBTV (see Figure 1). To synchronize the tablet and the TV application, the data-broker is used. When the TV application requires some user input, it sends a corresponding request to the data-broker, which forwards this request to the tablet. The input form is then shown on the tablet. After the user has provided the requested input, the input is sent back to the data-broker and shown on the TV.

2.2 Detailed description of Observation and Tracking

2.2.1 Overview

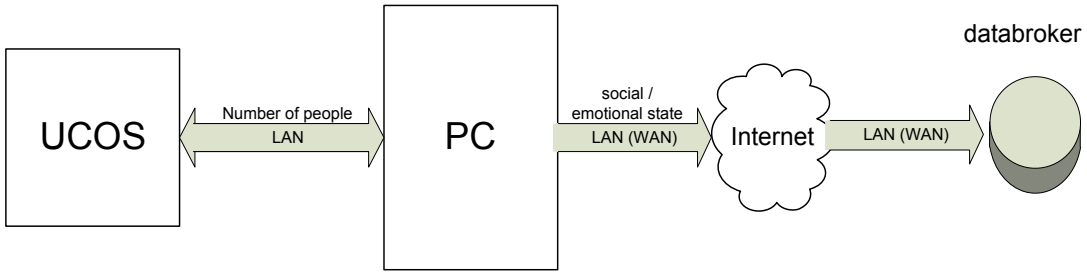

Figure 5: Block diagram of gesture recognition components

The Observation and Tracking function is utilized to estimate the number of persons located in front of the social media TV. This observation together with a time-stamp is transferred in real time via Ethernet UDP connection to the media PC and is analysed by dedicated modules in the gesture recognition software. The analysis will utilize the information: number of people, time of day to recognize the habits of the user and thereby derive an estimation on the social and emotional state of the user. Typical scenarios are a user not getting up in the morning at the usual time, not receiving a visit at a usual time of the day or at the usual day of the week. A pre-condition for this function is that these social contact and main activities of the user happen in front of the TV-set and/or the field of view of the sensor.

The social states derived and sent to the databroker can be information like: did not get up in the morning, did not receive a visit on a usual day/time, has been alone for a longer time, etc.

2.2.2 Interfaces

2.2.2.1 Internal Interfaces

The components of the system communicate over several different interfaces. UCOS is connected to the PC via a conventional LAN Interface, which could be a built in Interface in the PC or an external USB to LAN Adapter (which will be used on the development system).

It is also possible to connect the sensor to a local network switch. The Protocol used to communicate with UCOS is UDP.

Table 1 - Internal Interfaces

Component	Mechanical Interface	Protocol
Ucos ←→ PC	LAN (RJ45)	UDP

2.2.2.2 External Interfaces

The whole observation and tracking system communicates with the databroker, the receiver of the social/emotional state information. Data which are meant for the databroker have to be transported to a Server in the World Wide Web. Therefore a WAN Interface is necessary. This can be LAN, WIFI, mobile broadband...

Table 2 - External Interfaces

Component	Mechanical Interface	Protocol
PC →databroker (Internet)	WAN (via LAN, etc.)	UDP

2.2.3 UCOS2 XL

2.2.3.1 Purpose and role of the component in the system

UCOS is a Sensor for person counting, its sends current number of persons moving in the field of view of the sensor to a PC. The Sensor is used as user-input device.

The smart eye UCOS - Universal Counting Sensor is a compact 3D sensor system, equipped with twospecial optical CMOS sensors. These are characterized by exclusively registering moving objects. This is achieved with an internal movement detection that is implemented inthe recording element. The sensor system does not perceive video images, instead edge information from the monitored scene is directly processed on a digital signal processor.

With the novel principle of the sensor recording elements, the sensor only registers the edgesrespectively outlines of moving objects.

The heart of the system is a novel optical sensor chip with integrated on-chip pre-processing. Thismeans that numerous calculations are already executed on the sensor chip itself. Edges of moving objects are detected with high precision over time and transmitted in the form of an address and time stamp of the respective pixel to the digital signal processor. The information generated by a pixel that detects a change in light intensity is called an address event (AE).

The UCOS2 XL sensor has a practical ranging limit of 3 meters.

2.2.3.2 UCOS2XL data Format

The data format of the UCOS stereo sensor is a UDL packet containing the number of people and time-stamp information together with auxiliary data in a XML-like structure as shown below:

```
<SVSPCOUNT>
        <SENSOR>sensor_1</SENSOR>
        <PERSON>2</PERSON>
        <POS>0</POS>
        <BOARD_TIME>20090126 01:07:27</BOARD_TIME>
</SVSPCOUNT>
```

The exact syntax of the XML data packets to be used in FoSIBLE will be specified later in the project.

2.3 Detailed Description of Gesture Recognition

2.3.1 Overview

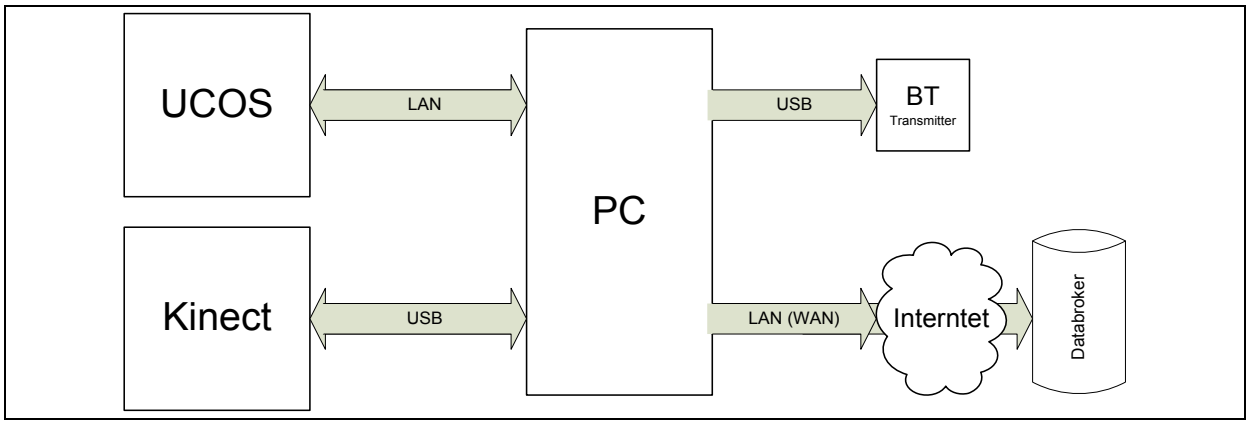

Figure 6 -Block diagram of gesture recognition components

The components of the gesture recognition System consists of three major parts. The primary parts are the sensors, which can be seen on the left side in Figure 6. The central Component is a PC which will be specified in detail for both the Development environment and the End-User environment. On the right side of Figure 6 the receivers of the gesture indicated commands are pictured. Some of the commands are used to control the functions on the TV-Set. A Bluetooth Transmitter can be seen in Figure 6 – this should only be uses if direct control via LAN is not possible. Some other commands or data packets will be sent directly to the external databroker, which is located on a server connected to the Internet.

2.3.2 Interfaces

2.3.2.1 Internal Interfaces

The components of the system communicate over several different interfaces. UCOS is connected to the PC via a conventional LAN Interface, which could be a built in Interface in

the PC or an external USB to LAN Adapter (which will be used on the development system). It is also possible to connect the sensor to a local network switch. The Protocol used to communicate with UCOS is UDP.

Kinect is connected to the PC via USB 2.0. For communication proprietary drivers are required, which are available for download. Driver informations are provided in §2.3.4.2.

Table 3 - Internal Interfaces

Component	Mechanical Interface	Protocol
Ucos ←→ PC	LAN (RJ45)	UDP
Kinect ←→ PC	USB 2.0	proprietary, drivers available

2.3.2.2 External Interfaces

The whole gesture recognition system communicates with the receivers of the Gesture commands over two different Interfaces. If required the optional Bluetooth/Infrared Transmitter is connected to the PC via USB drivers are available for download on the manufacturer's homepage. The Media PC used for FoSIBLE has a built in Bluetooth Module.

Commands which are meant for the databroker have to be transported to a Server in the World Wide Web. Therefore a WAN Interface is necessary. Other technologies such as LAN, WIFI, or mobile broadband might be used, as well.

Table 4 - External Interfaces

Component	Mechanical Interface	Protocol
PC →databroker (Internet)	WAN (via LAN, etc.)	UDP
PC → IR/BT Transmitter	USB 2.0	proprietary, drivers available

2.3.3 UCOS2 XL

2.3.3.1 Purpose and role of the component in the system

UCOS is a 3D Sensor, which detects motion near the sensor ("highest points"), tracks the motion and sends track information as address event data to the PC. The Sensor is used as user-input device.

All other specifications as in section 2.2.3 apply for the UCOS device.

2.3.3.2 UCOS2XL data Format for the FoSIBLE Project

The dataformat of the UCOS stereo sensor is a 2x32bit binary format encoding the event time and the pixel address (address-event representation) and the 3D depth, extended by some marking bits, in which tracking result of the sensor is encoded.

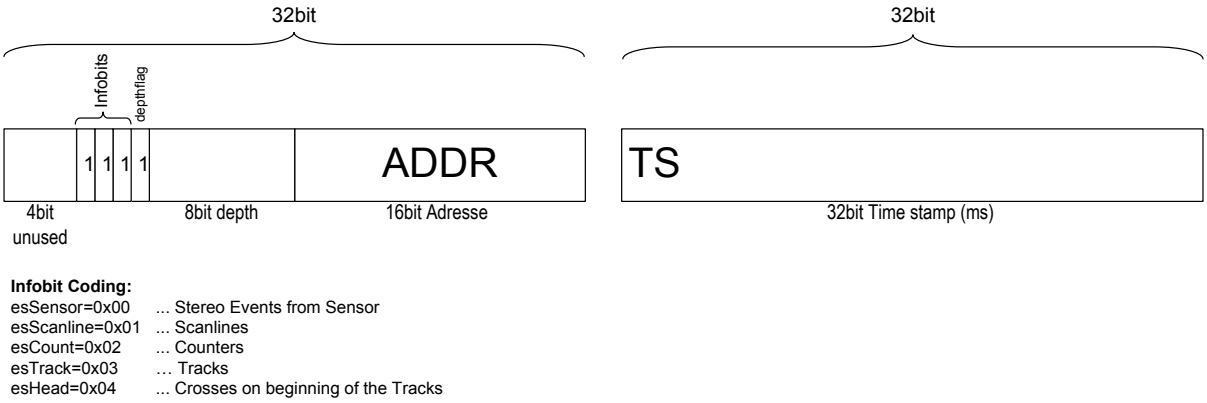

Figure 7 -UCOS2XL data format for FoSIBLE

Therefore a clear differentiation between depth Information, Tracks and other AE types is easily possible. Tracks and depth information will be used in the FoSIBLE Project. The track data represents the gesture made by the user.

2.3.4 Kinect

2.3.4.1 Purpose and role of the component in the system

Microsoft Kinect is a Sensor, which generates real time depth and color data of the watched scene. The Sensor is used as user-input device.

Microsoft Kinect uses the Primesense Reference Design for a Sensor called Primesensor. The depth sensor consists of an infrared laser projector combined with a monochrome CMOS sensor, which captures video data in 3D under any ambient light conditions. An IR-Pattern is projected to the scene and read back by the sensor. This read back is used by the sensor to calculate a depth image of the scene. This Technology is called “Light Coding”.

The Kinect sensor outputs video at a frame rate of 30 Hz. The RGB video stream uses 8-bit VGA resolution (640 × 480 pixels) with a Bayer color filter, while the monochrome depth sensing video stream is in VGA resolution (640 × 480 pixels) with 11-bit depth, which provides 2,048 levels of sensitivity. The Kinect sensor has a practical ranging limit of 0.7–6 m.

2.3.4.2 Implementation

The Kinect sensor is commercially available.

2.3.5 PC

2.3.5.1 Purpose and role of the component in the system

The PC is the central component of the system where all needed Software will be running and where all other components are connected to. See Appendix for detail.

2.3.5.2 Implementation of the End-User and Test Enviroment

The End-User and Test Environment is a MedionAkoya S4500D Multimedia PC which has Microsoft Windows 7 Home Edition 32-Bit installed. All required tools, drivers and applications will run on this System. The Hard- and Software is specified in **Table** 6 and **Table** 7.

2.3.6 Optional IR or Bluetooth Transmitter

2.3.6.1 Purpose and role of the component in the system

The IR or Bluetooth Transmitter is used to send commands to the TV Set. This could replace the standard IR Remote of the TV-Set to enable controlling some of the features of the TV-Set via gestures. This Component is only required if a direct control of the TV Set via LAN is not possible.

2.3.6.2 Implementation

A **U**niversal **I**nfrared **R**eceiver **T**ransmitter, called UIRT is used. It is connected via USB, a serial port is not required. The hardware and drivers are available at http://www.usbuirt.com/.

USB Bluetooth Dongles are available from various manufacturers.

3 Software Architecture

3.1 Gesture Recognition

3.1.1 SW Architecture Overview

A block diagram of the Software components for gesture recognition can be seen in Figure 8. The components marked green are part of the Development environment; all uncoloured parts are used in both, development and end-user environment.

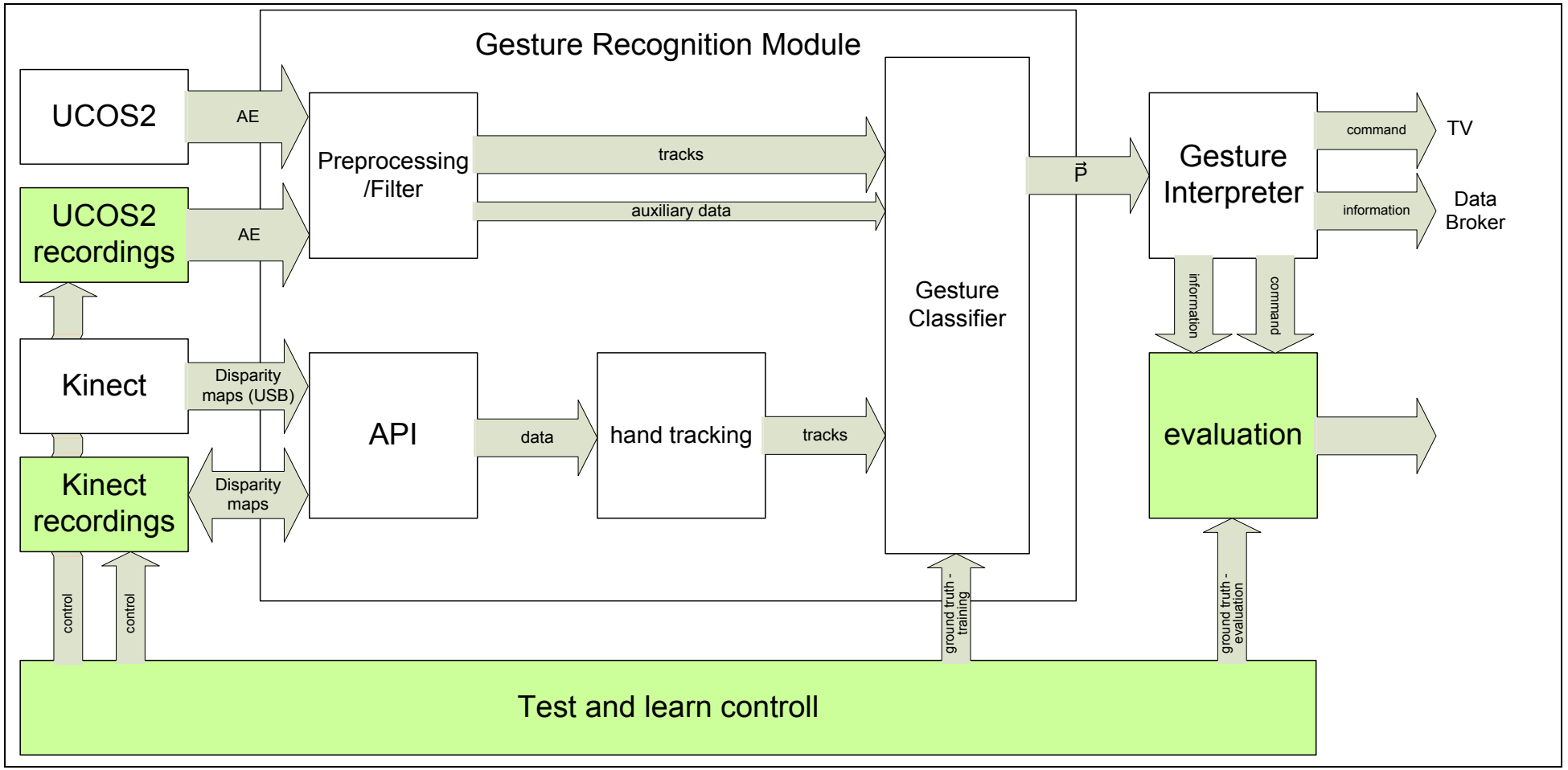

Figure 8: Block diagram of SW components and theirrelationship

3.1.2 Interfaces

3.1.2.1 Internal Interfaces

Table 5 - Internal Interfaces

Component	Protocol	dataformat
Ucos ←→ Preprocessing	UDP	AE (using AE Tracks)
Preprocessing ←→ Gesture Recognition		Tracks (x,y,d,t)
Gesture Recognition ←→ Gesture Interpreter/Evaluation		XML like Gesture Ids
Kinect ←→ API (Open NI od MS SDK)	USB 2.0	Frame based depth maps (images)
API (Open NI) ←→ Hand Tracking (NITE or MS SDK)		Frame based depth maps (images)
Hand Tracking (NITE or MS SDK) ←→ Gesture Recognition		Tracks (x,y,z)

3.1.2.2 External Interfaces

Component	Protocol	dataformat
Gesture Interpreter ←→databroker	UDP	XML

3.1.3 Preprocessing (Multistage)

3.1.3.1 Purpose and role of the component in the system

The data provided from the UCOS Sensor has to be filtered and smoothed. Unwanted information like noise should be removed and the paths of the detected movements should be refined. Preprocessing will implemented as a multistage chain of filters and smoothers. The data will be split up in Auxiliary information (derived from AEs not marked as track) and the tracks (marked as AE) detected by the sensor.

Auxiliary data should be the activity in a certain zone of view and other depth information received from the sensor.

The output data of the Preprocessing component are again tracks and auxiliary data, which will be used by the feature extraction units in the Gesture Classifier.

3.1.3.2 Implementation

For development the Preprocessor(s) will be implemented in Matlab. Later, for the UCOS device, the code will be migrated to the embedded system signal processor.

3.1.4 Gesture Classifier

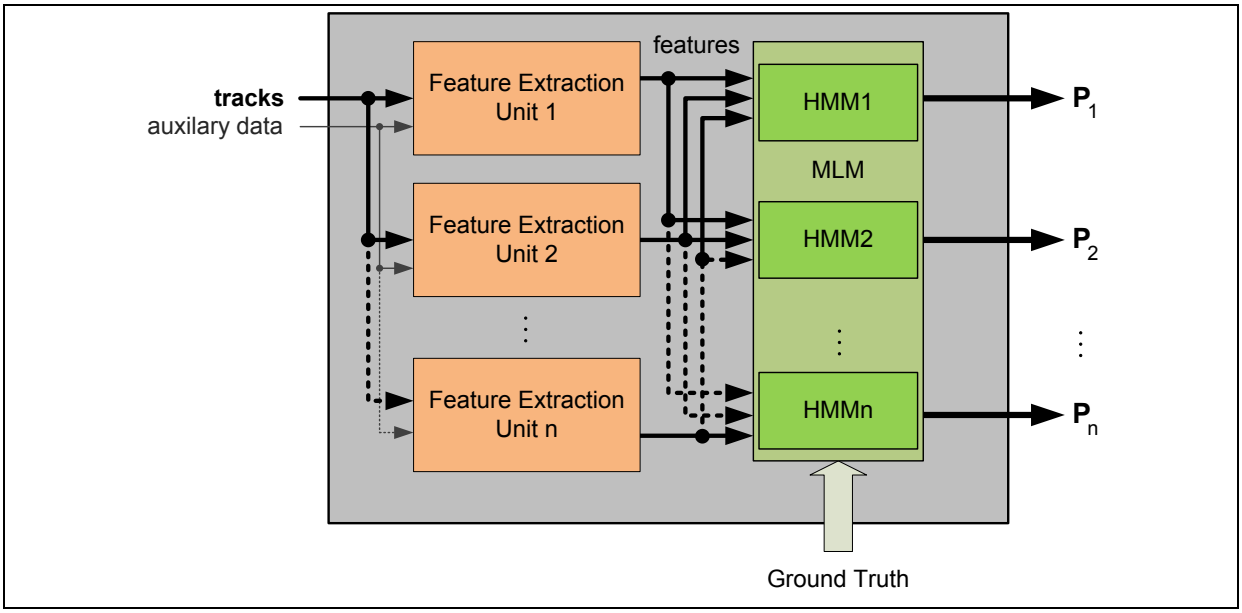

Figure 9 -Gesture Classifier

3.1.4.1 Purpose and role of the component in the system

The gesture classifier system consists of multiple feature extracting units and the classifier. The Feature extraction Units use the data received from pre-processing and derive explicit features like speed, acceleration, orientation, angle and activity.

These features are handed over to the Machine Learning Module (MLM), which will be based on a learning approach learning approach like Hidden Markov Models or Support Vector Machines.

The output of the MLM will be a Probability vector, containing the likelihood for each gesture.

For training purposes the Classifier will receive the ground truth information from the test and learn control.

3.1.4.2 Implementation

For development the gesture recognition will be implemented in Matlab.

3.1.5 Gesture Interpreter

3.1.5.1 Purpose and role of the component in the system

The Gesture commands can have different meanings. The Gesture Interpreter waits for gesture commands and translates the received information to data compatible for the receiver.

3.1.5.2 Implementation

For development the gesture recognition will be implemented in Matlab.

3.1.6 API

3.1.6.1 Purpose and role of the component in the system

The OpenNI Framework provides the interface for physical devices (Kinect) and for middleware components. The API enables modules to be registered in the OpenNI framework and used to produce sensory data.

3.1.6.2 Implementation

OpenNI is open Source and will be used as provided by the OpenNI organization.

3.1.7 Hand Tracking (NITE or MS SDK)

3.1.7.1 Purpose and role of the component in the system

Hand tracking is a part of the NITE framework provided by Primesense. After having performed a focus gesture (e.g. wave) the hand is tracked. The output is the hand‟s position in real-world coordinates in units of mm in each frame.

3.1.7.2 Implementation

The Algorithms for Hand tracking using the Kinect are provided by the Primesense Middleware NITE.

3.2 User-system interaction

To enhance the usability of the tablet application, the layout closely resembles the layout of the HBBTV application. As, however, the tablet only provides limited space for displaying content, each input field is shown individually on one screen. Thus, navigation to the previous and next input field was added. The general layout of this input forms is shown in Figure 10 and Figure 11.

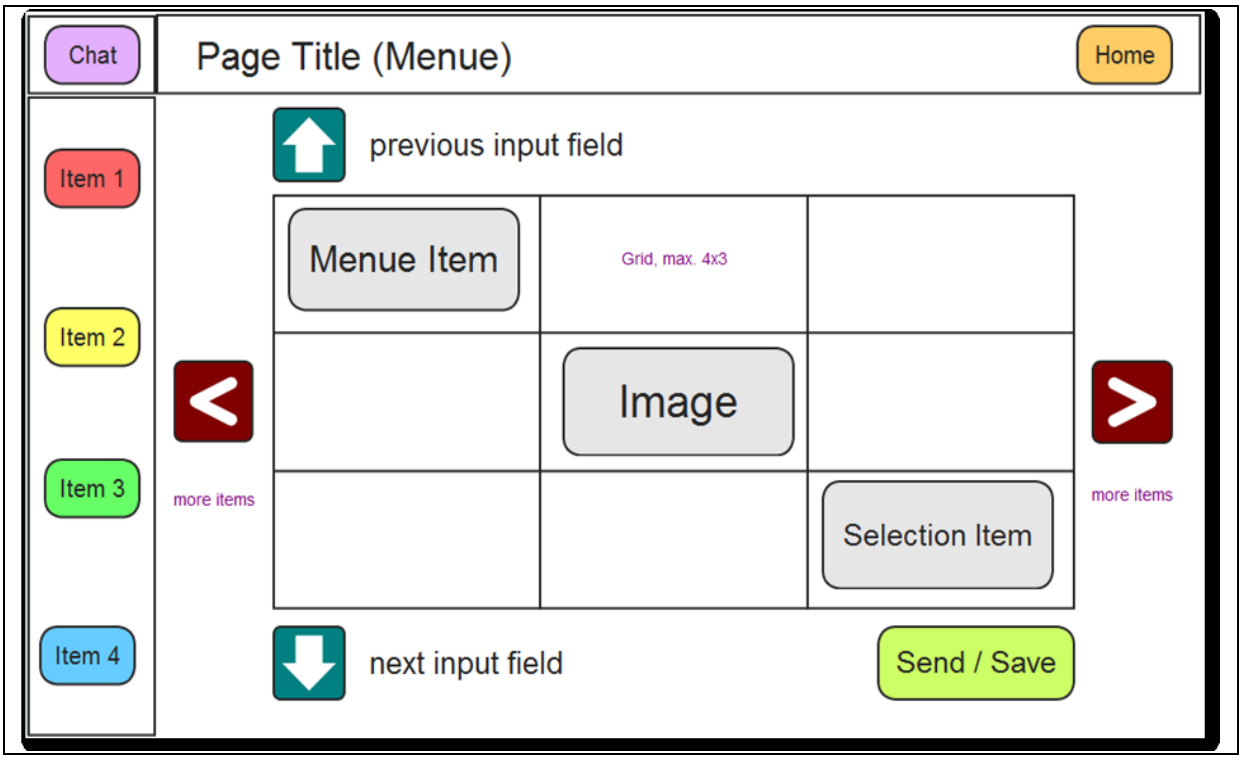

Figure 10 -Mock-up of aselectionform for the tablet

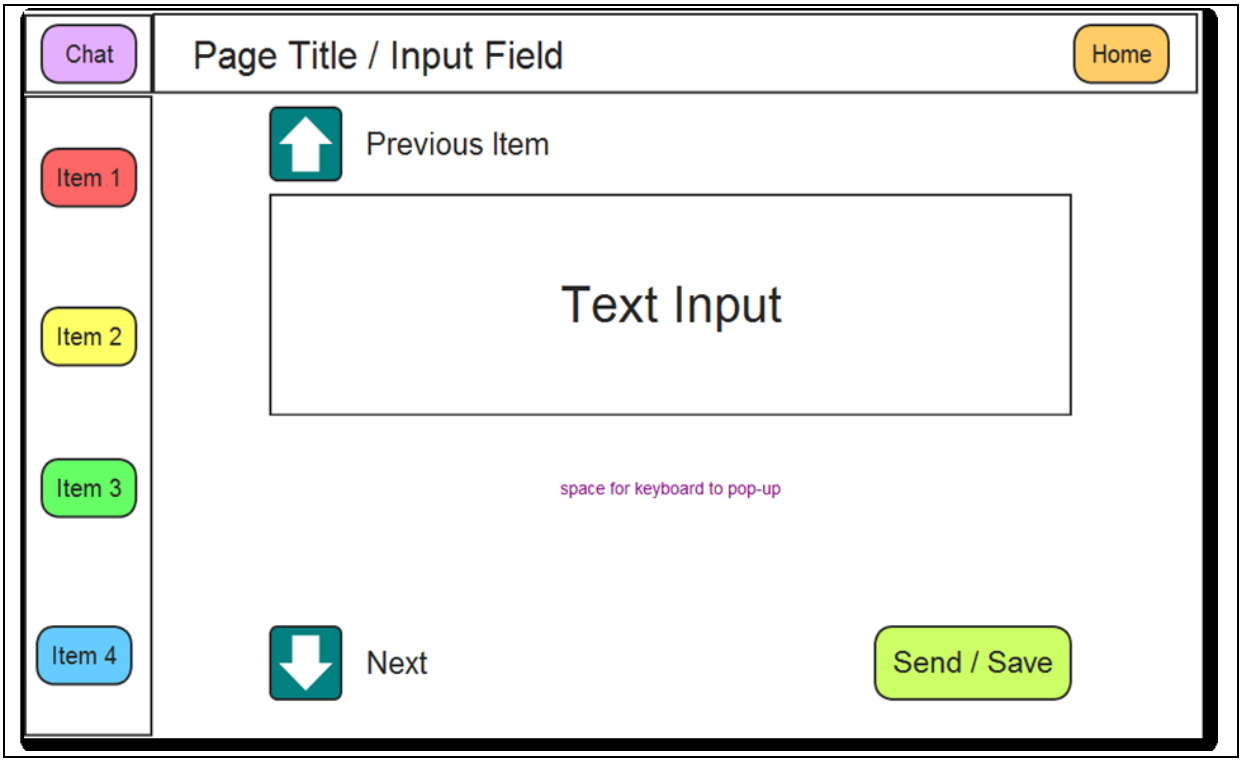

Figure 11 -Mock-up of an input form for the tablet

4 Evaluation Tools

4.1 Evaluation and Test Strategy

The strategy to evaluate the gesture recognition consists of several steps:

- Training- and test data recording
- Evaluation of the recognition rate
- Usability Tests
- Test operation in Living Lab Environment

4.2 Recording of Training- and Test Data

4.2.1 Recording tools

UCOS-smart eye center

The smart eye center SW tool will be used to record test-data and to setup the UCOS device. The smart eye center is a graphical user interface that allows to connect the UCOS device to a PC, configure the sensor parameters (such as sensitivity, thresholds, ..) and setup the device (IP address, netmask, etc.). It allows to display the live data stream of AER and the tracking data as well as to record sensor raw data and data playback. Figure 12 shows a screenshot of the smart eye Center SW. The smart eye Center SW runs under Windows XP and Windows 7 OS.

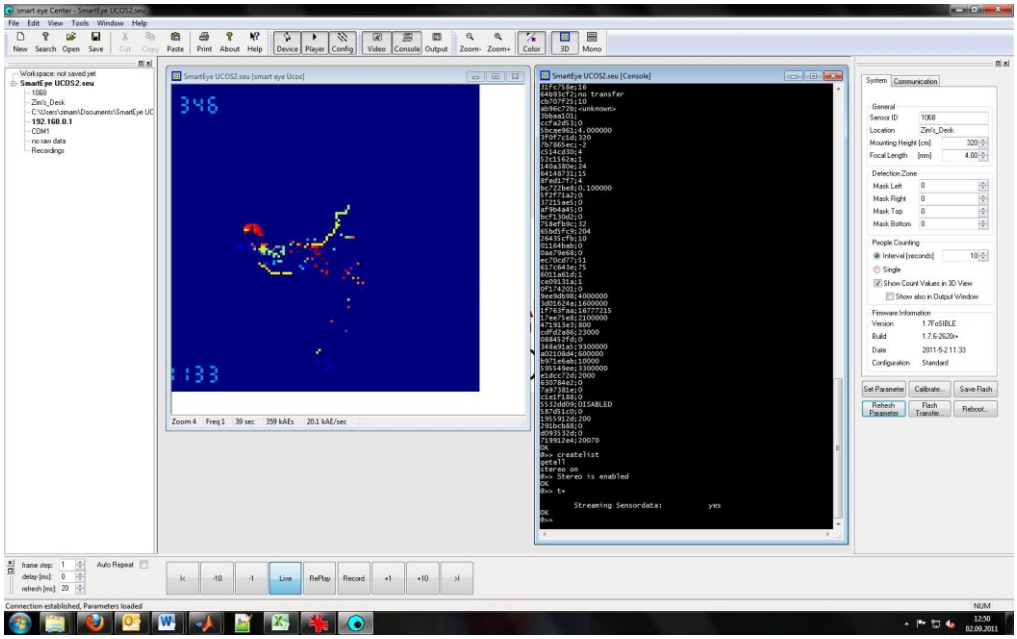

Figure 12 – smart eye Center tool for test data recording

Kinect data recording

Data from MS Kinect will initially be recorded and stored using Matlab. For the communication between Matlab and Kinect "Kinect-Mex" will be used, which is an open Source collection of wrapper functions of OpenNI APIs. No special tool boxes for Matlab are required. Recorded Data will be stored as a Matlab Structures and data files.

4.2.2 Test case selection and recording setup

For testing and learning data has to be recorded in a specific way. Each gesture has to be recorded several times. The Gestures should be stored in a way, that they could be used for automated learning and automated testing.

Recorded data will be split in training- and test data. All data will be annotated, i.e. the information what gesture is contained in the recorded data will be produced by human inspection. This results in so called ground truth data that are the basis for training and test.

Therefore each recording must have a unique ID, which makes it possible to assign it to more detailed information stored in a configuration/information file. In Figure 13 the setup for the Gesture recording can be seen. This mainly represents a typical living room situation (a person on a couch performing gestures in front of the TV Screen (and the sensor placed above the screen)), but more detailed information is required for evaluating the sensor performance. Distance d between sensor and person has to be stored, and also deviation of the gesturing position to centre position should be record as angle α.

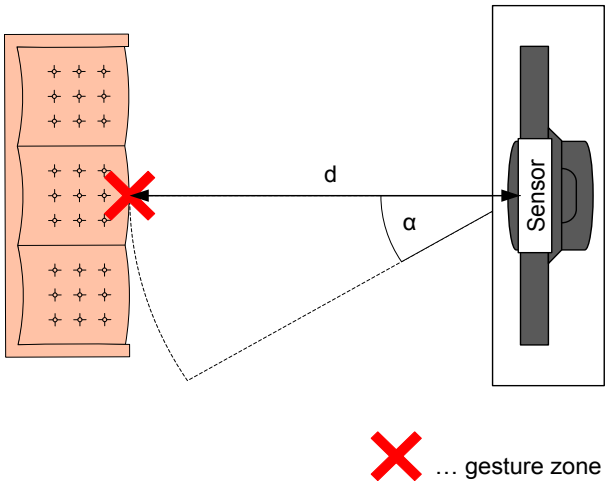

Figure 13 - GestureRecording Setup

Further Information that should be stored is Information about the person performing the gestures (age, sex, size…) and information about ambient conditions (light conditions…).

4.2.3 Dataformat

For each recording ground truth Information must exist. This information is used for evaluation and learning. Each occurrence of a gesture should be noted in a “Logfile”, which will be saved together with the recording.

Each Log entry will contain:

Timestamp; gestureID; distance[cm]; height [cm]; Angle[°]; LightConditions [dark,…]; personID

The Timestamp is the point in the recording, where a certain gesture (marked by the gestureID) occurs. All other information like distance, height, angle and all other information's are for evaluation purposes.

4.3 Evaluation of Recognition Rates

In the early prototypes the evaluation of the recognition rate will be done after the gesture classifier has been trained by the training data sets. The gesture classifier algorithms will then be off-line, low-level tested in a PC environment. The resulting output probabilities produced by the classifier will be compared to the ground truth. From these results detection rates will be calculated and visualized in confusion matrixes representing the performance of the gesture recognition.

These tests are module tests and do not yet include real-time behaviour and user interaction.

4.4 Usability Test

Gestures (manipulations and deictics) are generally considered to provide a very direct and natural way of interacting with a system. However, little research exists on how older people could benefit from gesture based interactions and if they meet the special needs and abilities of the elderly. Older people tend to have particular problems with the interaction of new technology because devices are not designed to accommodate their special needs [8]. To gain knowledge which kind of menu is most feasible for gesture control in FoSIBLE we conduct usability tests with representatives of our target group. The research questions are:

- Which kind of gesture control is the fastest for selection tasks and produces an error rate below 5%?
- Which kind of gesture control do older users prefer in terms of usability and joyfulness?
- Do older users accept gestures as input method?

In order to answer the research questions a comparative study with menu technique and amount of menu items as independent variables will be carried out. The results will be presented in D5.3.

Usability Tests will be repeated with the final developed real-time systems and integrated HW in WP6 (D6.3).

4.4.1 Test Setup

Subsequent to a literature analysis of simple gesture based menu interactions we selected four different types for testing them with older people: (i) static hand positions for cursor control, (ii)hand movement tracking for cursor control, (iii) hand strokes in a radial menu, and (iv) dial plate for a rotary menu. Every menu type will be implemented with a similar user interface with four or eight menu items respectively. As it is neither feasible nor necessary to implement all menu types completely we make use of the methodology of Wizard of Oz [5]. This means in this case that we show an accordant TV user interface for

every menu type to the invited test persons and give them visual feedback about their hand position. However, the initiation of the gesture recognition and the selection will be triggered by the test supervisor by hand. The prototypes will be created with the Kinect for Windows SDK[1] to realize the tracking and with Microsoft Expression Blend[2] for the creating of the user interface. The output device will be a 32" Samsung LCD TV with a refresh rate of 100 Hz.

We will invite sixteen right-handed test persons older than 65 years for conducting the tests. Every participant has to perform the same tasks for every menu type but in different order to exclude biases. An introduction to every technique and some practise time with each menu type will be provided prior to data collection. Performance will be measured through task completion time and the amount of selection errors. In addition the participants have to answer the standardized questionnaires AttrakDiff [2]and TAM3 [10] subsequent to the test to gain knowledge about the user experience and acceptance of the gesture based menu interactions.

4.5 Test Operation in Living Lab

Tests operations of the final developed real-time systems and integrated HW in the Living Lab will be performed in WP 6 (D6.2). These tests are system tests and will include real-time behaviour and user interaction.

5 Appendix

5.1 Configuration Information

Software Type	Name Version	Download Link (date)
Kinect Driver	avin2-SensorKinect Win32-V5.0.0	https://github.com/avin2/SensorKinect/archives/unstable (17.03.2011)
Kinect API	Open NI Win32-V1.0.0.25	http://www.openni.org/downloadfiles/openni-binaries/20-latest-unstable (17.03.2011)
Kinect	NITE	http://www.openni.org/downloadfiles/openni-compliant-

[1]http://research.microsoft.com/en-us/um/redmond/projects/kinectsdk/

[2]http://www.microsoft.com/expression/

Middleware	Win32-V1.3.0.18	middleware-binaries/33-latest-unstable (17.03.2011)
UCOS Control	Smart Eye Center V1.1.8 Release	

5.2 Development Environment

The development environment is a Dell Optiplex 780 standard Office PC which has Microsoft Windows 7 Enterprise 64-Bit installed. All required tools, drivers and applications will run on this System. The Hard- and Software is specified in **Table** 6 and **Table** 7.

Table 6 – Development PC Hardware Dell Optiplex 780

Component	Name	Details
CPU	Intel Core 2 Duo E8500	3,17GHz
Memory (RAM)		4,00GB
Graphics	Intel Q45/Q43	onboard, DVI, VGA
Network Interface 1	Intel 82567LM-3 Gigabit Adapter	onboard, RJ45
Network Interface 2	D-Link DUB-E100	USB-LAN Adapter, RJ45

Table 7 – Development PC Software and Drivers

Software Typ	Name	Version
Operating System	Microsoft Windows 7 Enterprise	64-Bit
	Microsoft Visual Studio 10 Professional	Version 10.30319.1
	Matlab R2010b	32-Bit, Version 7.11.0.584
Kinect Driver	avin2-SensorKinect	Win32-V5.0.0
Kinect API	Open NI	Win32-V1.0.0.25
Kinect Middleware	NITE	Win32-V1.3.0.18
UCOS Control	Smart Eye Center	V1.1.8 Release

5.3 End User PC Environment

Table 8 – End-User and Test PC Hardware, Medion Akoya S4500D

Component	Name	Details
CPU	Intel Core 2 Duo E7300	2,66GHz
Memory (RAM)		2,00GB
Graphics	NVIDIA Geforce 9300	onboard, DVI, VGA, HDMI
Network Interface 1	Gigabit Adapter	onboard, RJ45
Network Interface 2	D-Link DUB-E100	USB-LAN Adapter, RJ45

Table 9 – End-User and Test PC Software and Drivers

Software Type	Name	Version
Operating System	Microsoft Windows 7 Home Edition	32-Bit
Kinect Driver	avin2-SensorKinect	Win32-V5.0.0
Kinect API	Open NI	Win32-V1.0.0.25
Kinect Middleware	NITE	Win32-V1.3.0.18
UCOS Control	Smart Eye Center	V1.1.8 Release

5.4 Tablet PC technical specification

Table 10 – Tablet PC technical specifications

Manufacturer	Archos	
Product Name	70 internet tablet	
OS	Android V2.3	Version
Memory (RAM)	256	MB
Memory (Flash)	8	GB
CPU	ARM Cortex A8 at 1 GHz	
Display	7”, 800x480 pixels, 16 mio. Colors	

Network	WiFi 802.11b/g/n, Bluetooth 2.1 EDR	
Weight	300	gr

6 References

[1] Chertoff, D. B.; Byers, W. R.; LaViola Jr., J. J. (2009). An Exploration of Menu Techniques using a 3D Game Input Device. ICFDG (Vol. 83, pp. 256-263).

[2] Hassenzahl, M. (2006). Hedonic, emotional and experiential perspectives on product quality. In: C. Ghaoui (Ed.), Encyclopedia of Human Computer Interaction (pp. 266-272)

[3] Jetter, H.-christian, Gerken, J., & Reiterer, H. (2010). Natural User Interfaces : Why We Need Better Model-Worlds , Not Better Gestures. Natural user interfaces: the prospect and challenge of touch and gestural computing (Workshop CHI 2010) (pp. 1-4).

[4] Karam, M., & Schraefel, M. C. (2005). A taxonomy of Gestures in Human Computer Interaction. ACM Transactions on Computer-Human Interactions, 1-45.

[5] Kelley, J.F. (1984). An iterative design methodology for user-friendly natural language office information applications. ACM Transactions on Office Information Systems, March 1984, 2:1 (pp. 26–41)

[6] Kray, C., Nesbitt, D., Dawson, J., Rohs, M., & Laboratories, D. T. (2010). User-Defined Gestures for Connecting Mobile Phones , Public Displays , and Tabletops. Proceedings of the 12th international conference on Human computer interaction with mobile devices and services (pp. 239-248).

[7] Primesense, The PrimeSensor Reference Design 1.08 datasheet, Available at http://www.primesense.com/files/FMF_2.PDF as of 2011-03-15

[8] Stößel, C., Wandke, H., & Blessing, L. (2010). Gestural Interfaces for Elderly Users: Help or Hindrance? Gesture in Embodied Communication and Human-Computer Interaction, 269–280. Springer.

[9] UCOS2 data sheet - http://www.ait.ac.at/research-services/research-services-safety-security/new-sensor-technologies/development-of-embedded-systems-for-customer-specific-solutions/smart-eye-ucos-universal-counting-sensor/?L=1 as of 2011-08-31

[10] Venkatesh, V., and Bala, H. “Technology Acceptance Model 3 and a Research Agenda on Interventions,” Decision Sciences, 39, 2008, 273-315