Authors:
Louis Vuurpijl, Louis ten Bosch, Stéphane Rossignol, Andre Neumann, Ralf Engel, Norbert Pfleger

Date: 15th March 2004

COMIC Deliverable 3.3
Reports on Human Factors Experiments with simultaneous coordinated speech and pen input and fusion

Document history

Date	Editor	Explanation	Status
Mar ’03	Vuurpijl&Boves	set up first draft of D3.3.1	(internal draft)
Apr ’03	ten Bosch&Rossignol	details on recognizers	(internal draft)
Apr ’03	Pfleger&Engel	FUSION and NLP	(internal draft)
May ’03	Neumann, de Ruiter, Boves, Vuurpijl	specs HF experiments	(internal draft)
June 10 ’03	Rossignol&Vuurpijl	final editing D.3.3.1	(internal COMIC doc)
Jan ’04	Neumann&Vuurpijl&ten Bosch	new set up D3.3	(first draft)
Jan ’04	Rossignol&ten Bosch	first experimental results	(draft)
Feb ’04	Neumann	analysis questionnaire	(draft)
Feb ’04	Pfleger&Engel	analysis NLP & FUSION	(draft)
Feb ’04	All authors	final editing last results	(draft)
Feb ’04	Vuurpijl&ten Bosch	final editing first version	(first version)
Mar ’04	Boves&den Os&White	review	
Mar ’04	Vuurpijl&ten Bosch	final editing	final version D3.3

COMIC

Information sheet issued with Public COMIC Document 3.3

Title: Reports on Human Factors Experiments with simultaneous co-ordinated speech and pen input and fusion

Abstract: This document presents the results of an elaborate human factors study on multimodal interaction in bathroom design

Author: Louis Vuurpijl, Louis ten Bosch, Stéphane Rossignol, Andre Neumann, Ralf Engel, Norbert Pfleger

Reviewers: Els den Os, Lou Boves, Michael White

Project: COMIC

Project number: IST-2001-32311

Date: 15th March 2004

Distribution list

COMIC partners: All
External COMIC: PO, Reviewers

Public

Key words: Human Factors experiments; multimodal system design and evaluation

The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

Contents

Chapter 1

Introduction

> *They drag programmers into dark rooms, where they watch through one-way mirrors as hapless users struggle with their software. At first, the programmers suspect that the test subjects have brain damage. They cannot believe that any user could be so stupid as to not understand their program. Finally, after much painful observation, the programmers are forced to bow to empirical evidence. They admit that their interface design needs work...* [2]

Research in multimodal interaction tends to divide into two categories that have little in common. One field focuses on relatively simple applications, where users interact with some kind of map, or complete some kind of form using a combination of speech and pen for input. More often than not, the pen can only be used as a pointing device. For entering alphanumeric input with the pen, a soft keyboard must be used, or the user must write isolated characters in a dedicated field on the screen. Examples of projects in this category are SmartKom [12] and MUST [4]. The other category addresses virtual reality applications, where the user can move around freely, while the system interprets a limited lexicon of speech and gestures that are relevant for the completion of a specific task [1]. In COMIC we intend to narrow the gap between the two categories, by extending the input and output capabilities of an application in the first category. The experiments described in this document focus on the input side exclusively, exploring the suitability of pen input recognition, automated speech recognition, natural language processing and multimodal fusion.

Projects in multimodal interaction differ in yet another aspect. Many projects seem to limit themselves to investigating ways in which several input and output modalities can be combined in human-system interaction. The focus in these projects is on experiments with procedures to interpret multimodal input, and methods for rendering information in parallel output channels. Another category of projects aims at developing operational multimodal services, most typically in digital telecommunication networks. The latter category of investigations is - by necessity - focused on developing interfaces that can be implemented and maintained in a cost-effective manner, yet are easy to use for a broad range of customers. In COMIC we want to combine the two research strands. On the one hand, we investigate fundamental issues related to the combination of multiple simultaneous input and output channels. But at the same time we are committed to designing multimodal interaction that is commercially feasible, in the sense that services can be built on a standard platform, and at the same time easy and attractive to use for uninformed customers. To move one step beyond the conventional map and form filling applications, we address an architectural design task, instantiated in the form of bathroom design. The bathroom design application has speech and pen input recognition at the input side. The user is able to make sketches on a pen tablet for entering the shape and dimensions of a bathroom. In addition, users can point at objects on the screen, such as bathtubs, basins, faucets, etc., and ask the system to shown alternative designs.

The user interacts with the system in three phases. In the first phase, the user enters the ground plan of the room, containing walls, location and opening direction of doors, and the location of windows. Relevant measures such as wall lengths, window width and heights are entered as well. The ground plan containing these features determines feasible layouts of sanitary ware and additional bathroom furniture. After the ground plan of the room is entered, the system computes a number of alternative standard bathroom designs, which can be decorated with tiles and sanitary equipment in the second phase of interaction. Finally, the user can move through a 3D representation of the design, and discuss possible changes. Although all three phases are covered in COMIC, this report addresses the first phase of bathroom design: the process of entering the shape and size of the room. The focus of this report is on input recognition technology (pen input and speech recognition) and multimodal fusion.

1.1 Human Factors experiments in COMIC

COMIC aims to substantially advance our understanding of conversational and natural human computer interaction. More specifically, we pursue two closely related goals:

- acquire knowledge about the way in which humans interact with multimodal systems, and build formal models of that behavior that can be used to guide the development of future multimodal systems
- advance our understanding of how multimodal systems must be designed to be able to understand the user's intentions and to react appropriately.

Today, little observational data and even fewer formal models are available of human behavior and preferences in interacting with multimodal systems. Nevertheless, it is clear that human performance and appreciation is heavily influenced by the technical and functional capabilities of the multimodal systems that can be built with available technology that is currently available or that is under development in research laboratories. But then again, we do not really know which capabilities or limitations of the component technologies have the biggest impact in a specific type of application. Therefore, the Human Factors (HF) experiments in COMIC serve two general goals:

- they must improve our understanding of human preferences and limitations in the interaction with multimodal systems, and
- they must help us to pinpoint the bottlenecks in the performance of the systems and their components that need to be amended most urgently to accomplish real and substantial improvements in the perceived quality of these systems.

A corollary goal of the HF experiments in COMIC is the collection of data that can be used to train and improve the processing modules. This goal is pursued by recording and annotating all input produced by the subjects during the experiments.

1.1.1 Understanding human behavior

With respect to user behavior and user preferences in multimodal interaction we want to investigate how uninformed subjects go about entering information into a multimodal system when they can use both pen and speech. More specifically, we want to investigate under what conditions users prefer pen, speech, or some kind of combination of pen and speech, depending on the type of information that must be specified. It goes without saying that user behavior and user preferences will be affected by the recognition performance (in terms of speed and accuracy) of the input channels.

Because recognition performance is less than perfect (and will stay so for a long time to come) users are confronted with the need to detect and correct recognition errors. In ’natural’ human-human interaction, both parties are equally responsible for monitoring and maintaining the interaction. In human-system interaction, however, we are confronted with the inability of the present technology to reliably detect that a user input was probably incorrectly recognized. As a consequence, the responsibility for monitoring the progress of a dialog rests almost exclusively with the human partner in the interaction. In performing this duty, the human is completely dependent on the feedback that is provided by the system. Therefore, we cannot investigate the effects of recognition and understanding performance on the user's behavior other than via the system's response. In the tasks that we address in the context of the COMIC demonstrators (specifying the design of a bathroom) and especially in the first phase of a session between a user and the demonstrator system (during which the shape and dimensions of the room must be provided), a large part of the system's responses will be in the form of text or graphics.

1.1.2 Guiding technology development

In addition to better understanding user behavior and user preferences, the HF experiments play a decisive role in the improvement of the performance of the architecture and the component modules in the integrated 'COMIC system'. In the experiments reported here, the focus is on the Input modules that are responsible for recognizing the multimodal input from the user, and the Fusion module that integrates and interprets the information from the various input recognizers. To that end, the HF experiments will provide a collection of realistic and useful field data. An in-depth analysis of the recognition performance will pinpoint the problems that have the largest impact on the behavior and the appreciation of the subjects. In addition, these data are essential for training the recognizers and the fusion module, and thus for improving the performance of the overall system.

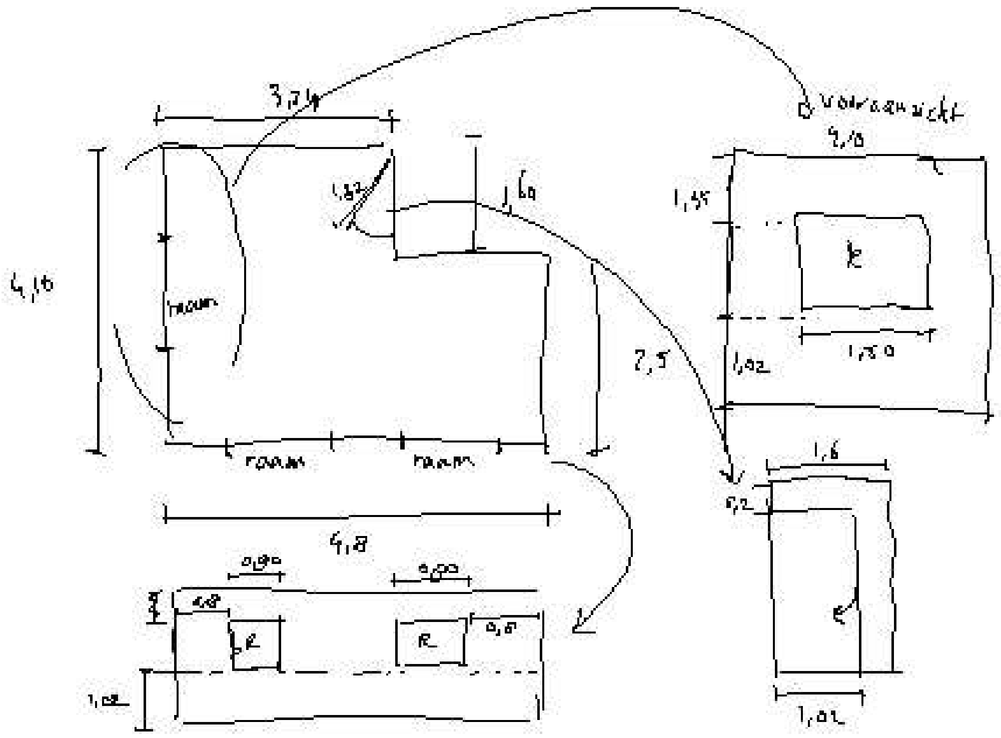
1.2 Earlier results

In [9], two pilot studies are described that explore multimodal interaction in the context of bathroom design. The first pilot served as a feasibility study for Phase-I of the COMIC demonstrator, in which the layout and dimensions of a bathroom have to be specified. More specifically, two classes of information had to be entered by each subject: (i) the length and width of the room, position of window(s) and door(s), and (ii) bathroom furniture (i.e., toilet, bath, shower, sink, mirror, cupboard, heating, etcetera). The subjects (7 in total) were asked to imagine that they were communicating with a computer that can understand pen and speech input. The experimenter told the subject for each of the nine blueprints which mode (either pen only, speech only, or both) had to be used. At the end of the session, subjects had to fill in a questionnaire, reporting about how they experienced the use of speech only, pen only or the multimodal case. The task of each subject was to study a number of bathroom blueprints and subsequently copy-draw these by use of pen and speech, in a way that a human experimenter would be able to understand the information. To confirm that understanding, the experimenter was allowed to ask directed questions about missing or ambiguous information.

The main results of the first pilot study can be summarized as follows:

1. Subjects like the task. Although none of the subjects had previous experience with a tablet computer and speech recognition, they were able to complete the tasks.
2. One goal of the pilot experiments was to get insight in how users prefer to enter specific types of information. From the questionnaires it appeared that subjects did not like the "speech only" mode. The "pen only" mode was rated much more favorably, but overall, subjects preferred the combination of pen and speech. All subjects preferred the pen as an input modality for drawing physical elements such as walls, windows and doors, and half of them preferred the speech mode for entering measures or other additional information.
3. The variation between subjects is very high, in particular if they are left free in their interaction. Depending on the artistic skills, some subjects may draw extremely complex sketches that pose difficulties for the recognition of their blueprints (see, e.g., Figure 1.1). Three classes of pen-based gesture and handwriting input were found: (i) graphical input for the specification of walls, windows, doors and bathroom interior objects, (ii) character and word input for the specifications of lengths and widths (units) and annotations of the drawings and (iii) digits for the numerical specification of lengths, widths and heights. A fourth class of pen input comprised deictic references (pointing gestures, area markings) and erasure commands (by using the back of the pen).

Figure 1.1: The challenge in gesture recognition: graphical drawing primitives, words, characters, digits, deictic references, and textual annotations.



4. For speech, a large inter-subject variation in dealing with the lay-out of the room map and with interior details was observed as well. Three "layers of use" can be distinguished: (i) thinking aloud ("I think I would place this over here"), which is not to be confused with self-addressed mumbling. Mumbling did not occur frequently, and subjects were explicitly asked to speak as clearly as possible; (ii) implicit question for approval from the experimental manager ("I hope this will be OK"), and (iii) real input to the "interpreting computer": factual ("Here a bath."), and of previewing type ("Now I will...").

5. About the relation between gestures and speech, it can be tentatively concluded that a pen gesture act is associated with a speech act (= an associated sequence of utterances) much more often than the other way round. One pen gesture "icon" usually goes with quite an amount of speech. It is not straightforward to divide the incoming speech into separate speech acts on the basis of the alternation of speech and silence only. However, when the gesture information is available, relations between a group of gestures on the one hand and a group of utterances on the other hand are clearly distinguishable.

In the second pilot study, it was examined whether subjects could deal with system-driven dialog strategies in the context of bathroom design. Eleven subjects took part in this study. Subjects had to follow precise instructions (from a human wizard), using speech and/or pen to enter information about the shape of bathroom and lengths of walls, position of doors, their widths and ways they open, position of windows, their widths and height from the floor, and position of drainage pipes and objects to which they are linked, e.g. bath, basin or toilet. Each subject had to provide the requested information by the use of a pen on a graphical tablet or by speech. In one condition, the instructions of the system pertained to atomic pieces of information (e.g., "draw one wall"; "enter the length of this wall", etc.). In the other condition the pieces of information prompted for by the system were more complex, thereby leaving more freedom to the subject to express themselves. For example, the system would give instructions like "enter the shape or circumference of the room", or "give the lengths of the walls". To specify the shape of the room the subjects could draw four or six lines, or a single closed contour, or any combination of lines they thought would be most appropriate.

Although the subjects did not like the system driven interaction very much, there are a number of reasons -explained in the next chapter- why we decided to use system driven interaction in the first version of the operational COMIC system.

1.3 Research goals

Given the state-of-the-art in the field of multimodal interaction, combined with the status of the development of the COMIC demonstrator for bathroom design, the HF experiments with the current system can only be exploratory in nature. The general set-up of the experiments is follows: subjects carry out a number of tasks interacting with the system, which is controlled by a human wizard. All actions and utterances of the subjects and the system are logged and annotated. This provides the 'objective' data in the experiment. Logged data comprises audio, video, and pen coordinate recordings, as well as system loggings. At the end of the session, each subject is requested to complete a questionnaire, consisting of a number of Likert scales. The responses to this questionnaire provide the subjective data in our experiment. A large number of interesting research questions can be explored by searching for relations between the objective and subjective measurement data. The results of this analysis will enable us to formulate more precise questions, which will be addressed in upcoming experiments. They will also enable us to pinpoint those performance issues that appear to affect the appreciation and the behavior of the subjects most.

Despite the fact that the human factors experiments have an exploratory nature, the input recognition systems and multimodal fusion module have evolved during the past year of COMIC. This has made it possible to design an experiment that involves true automated interpretation of pen- and speech-based user inputs. By analyzing the multimodal interaction loggings, subjective user measures are compared to the (objective) system performances. As the analysis of multimodal data requires a huge effort from the analyzers for this purpose a dedicated tool was developed [11].

The research goals for the present study read:

- study the behavior of subjects as a function of system performance
- explore the acquired data on insufficiencies in the system
- assess the usability of the system by analyzing subjective experimental data
- translate human factor observations into directions for effective module improvements

The remainder of this document is organized as follows. In the next chapter, the dialog design and turn-taking mechanism are discussed. In Chapter 3, the design of the experiments is described. Also, the system that was used to conduct the experiment is described, as is the tool that was developed to support the processing of the data recorded during the experiment. Chapter 4 presents the results of the human factors study, focusing on the performance of individual modules, the behavior of subjects while interacting with the system and the evaluation of the subjects' appreciation of the system.
This report has two appendices. Appendix A contains the instructions and questionnaire that were used for the experiment. Appendix B provides a detailed specification of the technical capabilities of the input recognition and fusion modules. That Appendix also goes into detail with respect to the research questions for the individual modules.

Chapter 2

System-driven dialog and turn-taking

From the pilot experiments it appeared that, when left completely free, subjects who have no training in architectural design tend to draw sketches that are extremely difficult to interpret, even for humans (cf. Figure 1.1). The same holds for the speech they produce and that accompanies their drawings. This confirms earlier findings reported by Lee [5]. To avoid insurmountable problems for the input recognizers, it was decided to design a system-driven dialog, since a system-driven design narrows down the set of cooperative user dialog acts and avoids large numbers of out-of-domain or out-of-dialog speech and pen gestures. Following Oviatt [7] and Mankoff and Abowd [6], error prevention was applied by choosing system prompts that invite the subject to use restricted speech rather than spontaneous and natural speech.

2.1 Wizard-controlled system-driven dialogs

In COMIC, the module that is responsible for maintaining the state of the dialog is the Dialog Action Manager (DAM). Because at the time of designing and implementing the system for the HF experiments no such module was available, it was decided to develop a simple, yet effective, DAM for phase-I of the demonstrator. For entering the shape and dimensions of the room, the DAM can be implemented as a finite state machine, with state transitions that are completely determined by the interpretation of the most recent user input. Turn-taking is controlled by a strict protocol that prevents overlap between system and user dialog acts. In each dialog state the system starts with a prompt for information, after which the turn is passed to the user. Upon a user reply, the system interprets the provided information and decides which state transition has to be made. In the next state, the system utters the next prompt. This process is repeated until the system has received all required information.

The turn-taking protocol that is implemented in the system for the HF experiments allows the system to come back with a prompt after a certain time has been elapsed, also in the case of absent or non-interpretable replies from the user. This protocol, that is admittedly quite rigid, helps to avoid unduly long latencies. The protocol by which a turn is defined is described in detail in Section 2.4.

We employed a semi-automatic Wizard-of-Oz set-up for the human factor experiments. The human wizard is situated in a second room next to the experimental room, and the subjects were made to believe that they are interacting with a fully automatic system, to avoid them suspecting a Wizard-of-Oz set-up. At each dialog state, the automated DAM provides the wizard with a user-interface via which the wizard decides which feedback to generate and which state transition to take after each user input. A special tool was developed for this purpose, called the wizard support tool (explained in the next section). The wizard can make this decision based on the observed dialog acts (hearing speech, seeing pen trajectories, and watching the video signal).

Furthermore, the DAM processes the interpretations made by the system and (based on these), configures the wizard support tool in such a way that the system response is easily conceivable for the wizard.

For system output, the wizard selects and triggers pre-recorded utterances to instruct the subject how to proceed with the task. The wizard also triggers the rendering of beautification, including the erasure of beautified output in case the subject indicates that a recognition error has occurred. The possibilities for the wizard are the following:

- proceed to the next state,
- return to the previous state, in case of a recognition error detected by the subject.
  In this case the rendered system output is erased and the previous prompt is repeated, possibly in a more elaborate and specific manner.
- stay in the same state, if the system is not able to interpret the user input.
  In this case the previous prompt is repeated, possibly in a more elaborate and specific manner.

It should be noted here that in many transitions between dialog states, the human wizard will merely follow the directions suggested by the DAM. In most transitions, a fully automated DAM could have been used, requiring no human wizard at all. On the other hand, certain dialog transitions have been identified where the control of a human wizard was required. In particular, cases where the user was requested to respond to yes/no questions and where the system was not able to correctly understand the user response required human interference. For example, one of the confirmations concerned the question whether the user was finished with entering information. If the user would reply negatively but the system would understand a positive response, the DAM would switch to another phase of the dialog from which no return would be possible.

2.2 Wizard support

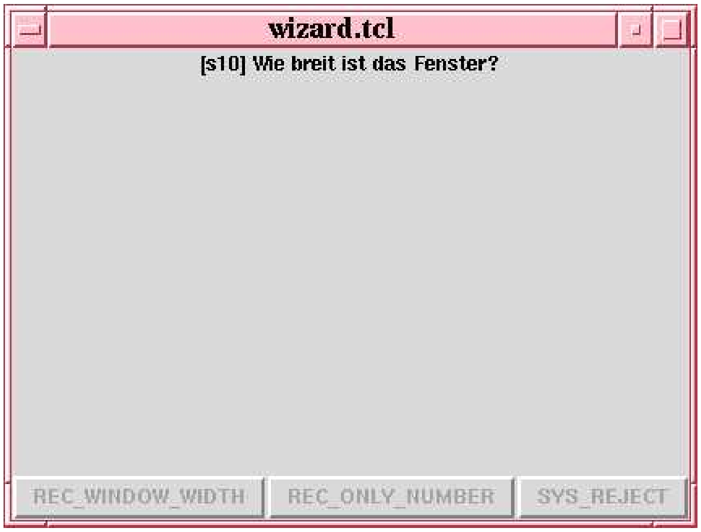
The task of the wizard is facilitated by a powerful and flexible wizard-support tool. Because the interaction is system driven, the wizard must follow the state transitions specified by the dialog description.

The FUSION module outputs semantically well-defined interpretations of the recognized speech and pen-input signals. However, as the output format is not easy to interpret for the human wizard, a tool is devised which supports the wizard in deciding which system responses to take and which state transition should be chosen. Based on the interpreted FUSION commands, the wizard is able to choose:

- Synthesized voice feedback: the wizard is able to emulate system prompts and feedback by choosing between the pre-recorded audio files, which are recorded by a native German speaking male person.
- Beautification: rendering of graphical objects, e.g. straight lines for walls.
- Texts: system feedback about the recognized pen and speech utterances (e.g., lengths, widths, heights) produced by the user.

The wizard support tool is implemented in Tcl/Tk and communicates with the DAM. Upon state changes, the DAM sends information to the tool, based on which it dynamically depicts the prompt that was uttered by the system. Furthermore, for each dialog state, the tool generates dedicated buttons via which the wizard can decide to switch to a specific next state. Below, an example screen shot of the tool is depicted. In state “s10”, the user is requested to enter the width of a window. Based on his/her input, the system can (i) either recognize a full dimension (comprising both length and unit), (ii) a length only, or (iii) it can reject the user input. The human wizard can observe the depicted prompts (indicating the expected user input), the coordinates displayed on the pen tablet, the audio and video channels, to support his decision. In this particular example, the wizard support tool receives three buttons from the DAM. Each button corresponds to one of the three possible system interpretations and is labeled with the corresponding text label.

Figure 2.1: The wizard support tool



[s10] Wie breit ist das Fenster?

2.3 Reject versus implicit confirmation

The possibility for the human subject to reject a certain system interpretation is one of the key issues in a dialog system. In this experiment, we used the strategy of implicit confirmation. The subject can only reject the interpretation of the system in the turn immediately following the turn in which it was rendered. To reject an interpretation, subjects can use a spoken utterance or a dedicated pen gesture (i.e., erase the last input). If the subject continues with the input for the new prompt, the system will tacitly assume that the previous recognition result was correct. Therefore, the system will freeze the previously rendered information and the subject will not be able to change the recognized value at a later stage in the interaction. If a subject wants to reject a recognition, rendered in the form of a beautification, he/she can use the pen to erase the input or say some negation phrase such as “No, this is not correct”. The system will not ask for an explicit acknowledgment of the recognition results. In the debriefing interview with the subjects we inquired whether subjects have inferred and understood this implicit confirmation strategy and its implications. All users confirmed to this inquiry, from which we conclude that implicit confirmation is a usable alternative to the set up where the system explicitly asks the user whether a turn is completed.

2.4 Turn-taking in COMIC

A schematic overview of the turn-taking protocol is depicted in Figure 2.2.

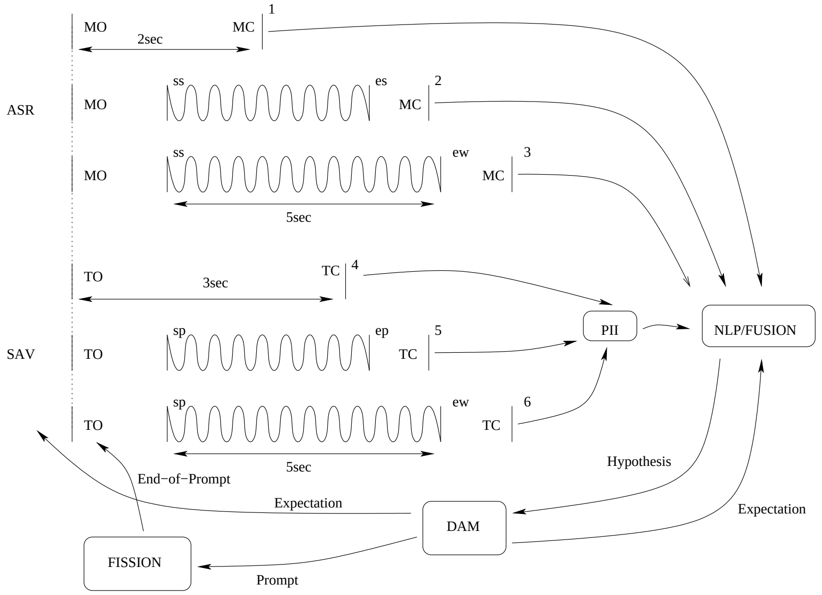
In Figure 2.2 the input channels “microphone” and “tablet” are controlled by the ASR and SAV[1], respectively. Initially, both channels are closed, which is required for obtaining a strict turn-taking protocol, where the system and the user cannot speak at the same time.

2.4.1 Definition of turn-taking: turn-pairs

Ensuring that the system and user cannot simultaneously obtain the turn is implemented in the current system as follows. Initially, the system has the turn. The DAM evokes a pre-recorded prompt. After the prompt is finished, DAM issues an expectation. The expectation tells the ASR, SAV, PII and FUSION modules that the

[1] SAV stands for Service Application ViSoft, which will be responsible for capturing pen input in the T24 COMIC system

Figure 2.2: Turn-taking protocol for the HF research platform. MO=microphone open, MC=microphone closed, TO=tablet open, TC=tablet closed.



system turn is finished and what the user is supposed to do next. Once SAV and ASR receive the expectation, they open the tablet and microphone and the user has the floor. Upon opening the tablet, SAV displays a green rectangle in the upper-left corner of the display. As long as the color of this rectangle stays green, the user is aware of the fact that he is allowed to speak or produce ink. SAV and ASR either send recognition results or a `recording_stopped` message to FUSION, which transmits this to the DAM. Upon closing both input channels, the green rectangle is marked red again and the user turn is ended.

2.4.2 How does turn-taking work?

For both SAV and ASR, three situations may occur:

1. The user keeps silent.
   Once an expectation is received by SAV and ASR, they open the tablet and microphone (TO and MO in Fig. 2.2) and start listening for a certain time interval (three seconds for SAV, two seconds for ASR). The two seconds time window during which the subject can start speaking is standard in spoken dialog systems. Initial experiments showed that subjects often took longer to start writing/drawing. Therefore, the user is given three seconds to start pen input. It is known that the cognitive functions that trigger handwriting require more time than when evoking speech. If no input is seen during these time windows, the input channels are closed and both decoders send a message to NLP/FUSION stating the user said/drew nothing. This situation is reflected by the numbers 1 and 4 in Fig. 2.2.

2. The user generates some input within the two (or three) seconds time window (reflected by 2 and 5 in Fig. 2.2). In case of speech input, ASR will send a start-of-speech message to FUSION (ss in the figure). In case of pen input, SAV will start sending coordinates to PII, which will send a start-of-pen message (sp in the figure) immediately upon receipt of the first sample, to FUSION. In this situation it is assumed that an end-of-pen (ep) and an end-of-speech (es) are detected within a time window of five seconds following the start-of-input. After detecting es, ASR sends the recognized results to NLP/FUSION. After detecting ep, SAV sends a `do_recognize` command to PII, which will subsequently send the interpreted pen input to FUSION. Both ASR and SAV close their input channels after detecting end-of-input.

3. The same as situation (2), but in this case no ep or es is detected, as the user keeps generating input, exceeding the limits of the time window (3 and 6 in the figure). An end-of-window (ew) event is detected and all input acquired until then is recognized, after which the results are sent to NLP/FUSION in case of speech input, and/or to FUSION in case of pen input. Both input channels are closed.

Restricting the time the user is allowed to speak/draw is quite common in speech-driven dialog systems, where in many occasions the system will confuse background noise for user input, prohibiting the detection of an end-of-speech event. For tablet input, this situation is less likely to occur, basically only in case of non-cooperative users or in cases where the user lays the pen down on the tablet.

Once FUSION received messages from both decoders, it is guaranteed that no more user input can be expected, as both channels are closed. The end of the user turn is guaranteed and FUSION/DAM can now process the information (which can be void) to determine the next system turn.

As mentioned above, a number of interactive experiments were performed in order to assess the usability of the timing parameters. The current human factors experiments are used to examine whether two seconds microphone opening time and three seconds tablet opening time are suitable. Similarly, the period of five seconds during which users can produce information will be assessed on its suitability.

Chapter 3

Implementation of the human factor experiments

This chapter describes the design of the human-factor experiments and the way in which the acquired data are analyzed. The details of the task are described, followed by an account of equipment, system and software, data, and choice of subjects. The last section describes our novel data analysis method, which is targeted on the research area of multimodal dialog evaluation.

3.1 Description of the task

The tasks that subjects have to perform in Phase-I of the COMIC demonstrator is the specification of bathroom layouts and dimensions. Although a bathroom can have almost any arbitrary shape, it appears that in actual practice shapes that differ from either a rectangle or some form of ’L’ are extremely rare. Therefore, in the current experiments, subjects are asked to specify the layout of a bathroom constrained to a rectangular shape. After they have provided input about the shape and dimensions of the bathroom, subjects have to indicate the position of doors and windows. This means that we have to distinguish six categories of information that a user can provide. These categories are: (i) wall information, (ii) door information, (iii) window information, (iv) dimensions, e.g. lengths, widths, heights of bathroom objects, (v) affirmations and negations, i.e. replies to yes-no questions like whether there is a door, a window, another door etc., and (vi) exceptions, i.e. in cases where a subject wants to repair system errors. Most of the information items in these categories comprise more than one atomic piece of information. The atoms, which form the basis for the design of the input recognizers, will be explained in more detail below.

The HF experiments are divided in four phases. At the start of the experiment, the experimenter will explain the task to the subjects. In this first phase, the experimenter explains the environment, the head-mounted microphone and the LCD-tablet with digitizing pen. The use of the eraser function of the pen is explained as well.

In the second phase, users are requested to draw three bathrooms from memory in an unconstrained, free condition. They are asked to enter the blueprint of their own bathroom, from a friend and from their parents. This phase does not involve automatic recognition and serves two aims. First, natural, unconstrained, dialog acts provide essential material to further develop the various modules in the COMIC system. Second, the subjects get acquainted with the task: using speech and pen to interact with a computer system. Just before the third phase, each subject is shown an instruction video containing examples of drawing with the pen and speaking to the system. In the third phase of the experiment, they have to copy the same data into a computer system, using the tablet to sketch and write, and using speech to provide information if subjects prefer to do so. Now, the computer does try to recognize all input gestures and utterances, using the system driven interaction strategy. After entering the data for the three bathrooms, subjects are requested to fill in a questionnaire in the fourth phase of the experiment, which is based on their experiences with interacting with the system during the third phase. For the remaining text of this document, we will focus on this third phase: the interaction of subjects with a system that performs recognition of their pen and speech input.

3.2 System-driven requests for information

As explained in the previous chapter, the system requests a subject for all necessary information in a number of distinct turns. Each turn is initiated by the system and starts with a prompt requesting for a piece of information. The following enumeration characterizes the objects that are requested:

- a wall
  to be sketched by the subject as something that can be interpreted as a line of a certain minimal length
- a dimension
  to be entered by the subject by writing and/or speaking. A dimension is a complex piece of information, in that it consists of an integer or rational number plus an optional unit indication. For the HF research platform, subject are able to write dimensions in several different forms, e.g. ”2.5 m”, ”2,5 m”, ”250 cm”, etc. Also, a range of spoken expressions is covered by the ASR grammar, including ”two meters fifty”, ”two and a half meters”, two hundred fifty centimeters”, etc.

  The syntactical processing of the output of the handwriting interpreter and speech recognizer is performed by the NLP component. For walls, the wall length is requested. For doors, standard widths are used, so no dimensions are requested. For windows, the height, width and height of the sill are requested.
- a door,
  which has to be entered by the subject by sketching and/or writing and/or speaking. A door is a complex piece of information, in that it comprises a wall in which it is mounted and a direction of opening (which in itself is complex, because it includes the position of the hinges and the direction of movement). It is up to the subject to choose a detailed drawing of a door, or to point to a location on a wall and to use speech to say that there is a door at that specific location.
- a window,
  for windows the same considerations hold as for doors, with the only addition that a window has a width, a height and a windowsill height, but no direction of opening.
- a confirmation (affirmation or negation),
  these are replies, semantically equivalent to a yes or no answer. For example, such requests are prompted when the system wants to know whether there is an additional window or wall. This type of user input will most likely be expressed by a spoken utterance.
- a correction,
  here, we mean an indication of an exceptional condition that requires correction, for example in case of recognition errors, or in case a subject wants to repair a mistake she/he made herself/himself. Subjects are told in the introduction that they can use the pen (by using the eraser function) and/or the speech mode to correct the faulty interpretations by the system. The basic form of a correcting phrase is a general negation, possibly followed by a positive corrective statement. Examples are:
  - **No, that is not correct.**
  - **No, this is not correct. This length is three meters forty.**
  - **No, the length is three and a half meters.**

  The phrases that can be recognized in this way are expected to cover most cases of error handling encountered during the previous human factor experiments. The advantage of allowing this type of correction is the relatively straightforward interpretation of these utterances by the NLP component. In the current implementation, simultaneously pointing (interpreted by the system as a confirmation) and uttering these negating phrases, is not allowed. However, we have not observed such combinations of pointing gestures and speech during any of the experiments.

The lexicon and language model of the speech recognition module have been adapted to cover these and similar utterances.

The information that the subjects had to enter in the system is requested in the order that is given in the following script:

1. Ask for SNR measurement. In order to calibrate the ASR system, subjects are requested to speak a number of sentences during four seconds. This information is required to estimate the signal-to-noise ratio.
2. Ask for a wall;
3. Ask for the wall length (steps 2+3 occur 4 times in total);
4. Confirm last size (switching to the entering of doors);
5. Ask for door;
6. If door opening missing, then ask for door opening;
7. Confirm door or door opening;
8. Ask whether more doors have to be entered. If yes, go back to step 5;
9. Else, ask for window location;
10. Ask for window width;
11. Ask for window height;
12. Ask for height of window sill;
13. Confirm latest input of height of window sill;
14. Ask whether more windows have to be entered. If yes, go back to step 9. If no, stop.

The confirmations from, e.g., items 7 and 13 are required to ask the user whether the information just entered was correct, after which the user is asked whether another objects needs to be entered. For example, this distinction is required to distinguish between a door being interpreted by the system as correct and the decision of the user about whether any more doors need to be specified. The minimal number of prompts is 4x2+1 for the walls and the corresponding wall lengths, plus the confirmation (items 2, 3 and 4); 3 for door, confirmation, and another door (items 5, 7 and 8); 4 for a window and corresponding dimensions (items 9-12), a final request for confirmation (item 13), and a request for another window (item 14). This amounts to 18 prompts and corresponding user responses, i.e., to 18 turns. Each session is preceded by an additional prompt-response pair for signal-to-noise ratio (SNR) calibration.

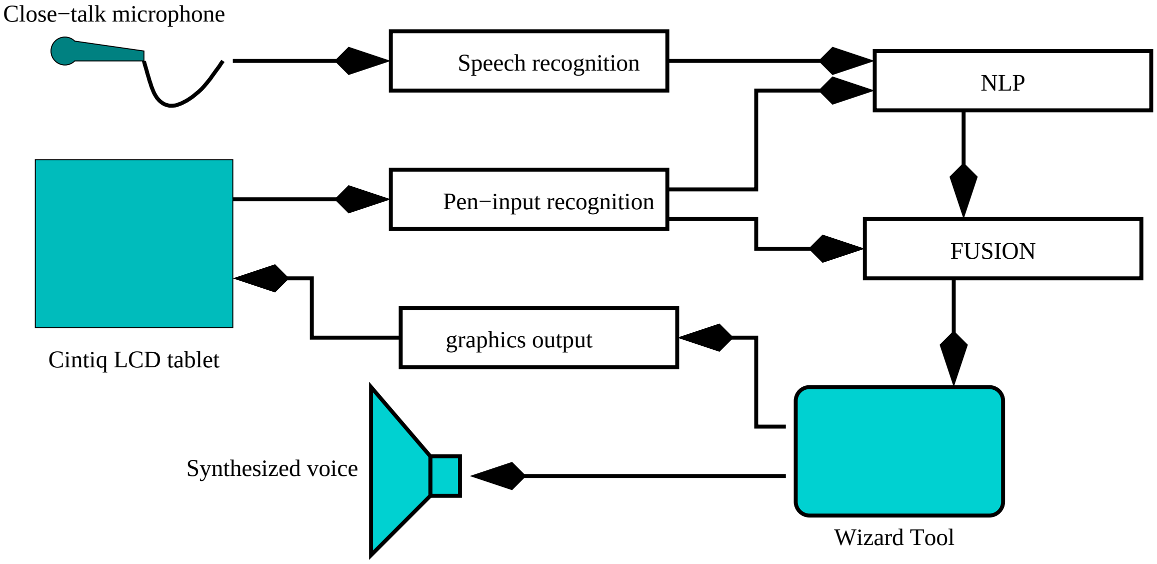
3.3 Equipment, system and software

Following the experiences with the SLOT paradigm (WP2) and our earlier experiments, the human factor experiments have been conducted using the following set-up:

1. Subjects use speech and pen to answer specific questions of the system.
2. The Wacom Cintiq 15X graphical LCD tablet is applied to acquire and render information from/to the subject. Acquired information is encoded in sequences of pen traces on the tablet. Rendered information is “ink” produced by a subject and textual and graphical feedback from the system.

3. Experiments are recorded on video, by using two camera's, one recording from above the tablet (recording horizontal movements), the other taking recordings from aside (recording vertical movements).
4. Subjects use a head-mounted close-talk microphone via which the speech signal is acquired.
5. The wizard is located in another room, interpreting the output from the FUSION module using the wizard support tool. The technical set-up of the experiment makes it possible for the wizard to see the screen and hear the speech from the subject. This allows the wizard to monitor all details of the interaction and override decisions made by the DAM in case of system interpretation errors.
6. All dialogs are logged with a level of detail that allows a meaningful post-hoc analysis. For each subject, the logging consists of two ASCII files and about one hundred audio files. It includes the timing of all events, the input, the input interpretations, the decisions of the wizard, and all beautifications. On the basis of this information, the data analysis can focus on the relation between observation data as a function of several factors: task-completion time, number of turns, amount of speech and/or pen-utterances, number of recognition errors, number of interpretation errors, number of user corrections.

Figure 3.1: System set-up for the HF experiments.



The experiments was run on two Linux machines. One system was used by the wizard to control the experiment and monitor recognized system states. The other was used to run the user interface, PII, ASR, NLP, and FUSION modules. Figure 3.1 depicts the set-up of the system.

3.4 Subjects

In the human factor experiments described in this report, 28 native German subjects took part. All were students in the University, or employees of the Max-Planck Institute of Psycholinguistics in Nijmegen. Most of these subjects had no technical training (Computer Science training), because we wanted to avoid a majority of subjects with a technical training. The questionnaire that subjects had to fill in contains the question: “How often have you used computer equipment the last five years?”, together with a 5 point Likert scale via which users indicated how often they have used a computer the last five years. The detailed instructions, as well as the questionnaire, can be found in Appendix A.
Of the 28 subjects 8 were male and 20 female. Their ages ranged between 20 and 37 (median 22 years). All but two of them were right-handed. Most of the subjects had a non-technical education and not much programming experience (average is 1.8). However, most reported to have quite some computer experience

(3.7), but very little experience using speech recognition systems (2.0) and even less using pen tablets (1.5). Subjects rated their artistic skills as high (3.6).

The system described in the previous chapter is the result of more than a year of programming, debugging, and extensive testing. It is beyond the scope of this report to provide details of all pilot experiments that have been performed, but the current functionality of the system could not have been obtained without the contributions of many subjects.

In addition to the 28 subjects in the eventual experiment, almost 30 other subjects were used in experiments aiming at debugging and fine tuning of the system. Part of these subjects were native speakers of Dutch with a good command of the German language. In addition, some of these subjects had experience with developing and testing multimodal interaction systems. The most important reason for using subjects for system debugging and fine tuning is the fact that it appears to be impossible to predict all reasonable behavior of uninformed subjects interacting with the system. In a similar vein, the messages that the individual modules in a multimodal system must be able to generate and to process can only be defined comprehensively by observing system failures in interaction with uninformed subjects. Thanks to the elaborate development work, we now have a system that is reasonably robust against unexpected behaviors of cooperative, but uninformed subjects.

In addition to completing the semantics of the modules in the system, the initial experiments were also decisive for tuning a number of essential parameters of the system. Specifically, experiments were needed to find suitable values of the timing parameters in the turn taking protocol.

3.5 System performance

In this section, it is described how for each of the modules relevant to WP3 and WP4, i.e., PII, ASR, NPL and FUSION, the logged data are analyzed. In order to label cases where modules were correct or made a wrong decision, a data annotation tool was designed. The tool is unique in that it is able to semi-automatically transcribe the recorded multimodal data, while using the pool loggings generated by the MULTIPLATFORM [3] communication mechanism to gather statistics about the performance of the individual modules. The tool is called μeval and is described in a paper to be presented at the LREC 2004 workshop on multimodal system evaluation. conference [11].

3.5.1 Analysis of multimodal experiment loggings

Previous research (e.g. [7, 8]) has shown that it is very difficult to make sense of the data recorded in multimodal interaction systems. Even if, as is the case in the present experiment, the interaction strategy is designed to constrain the user actions, multimodal interaction appears to offer many alternative ways to approach the goal. This large degree of freedom is especially important in the analysis of interactions with naive subjects, who lack the telepathic knowledge of the system In addition, objective data (the input and output of the individual modules in a system, including time stamps attached to actions of the system and the user) form a kind of cascade. In order to analyze the performance of individual modules the complete set of input and output messages must be considered. For speech and pen input this involves manual annotation of the physical input signals. Speech input must be transcribed verbatim, as well as in the form of the concept values expressed by the words. For pen input x,y,z coordinate streams must be annotated with the semantic labels that are relevant in the specific application. To assess the performance of modules that have no direct relations with physical input or output, such as FUSION, which receives symbolic input of the speech and pen input processors and passes symbolic data to the dialog action manager (DAM), input,output pairs must also be annotated for correctness (or type of error). In the past, the development of multimodal systems has been hindered by the absence of suitable tools for annotating and analyzing interaction data. For the analysis of the data acquired in the current HF experiments, we have used the analysis tool called “μeval”.

General structure of multimodal system loggings

Most communication platforms like Galaxy, the Open Agent Architecture and MULTIPLATFORM (muP) provide means to log system messages. Given the multi-modular nature of multimodal systems, and because modules are typically developed by different persons, system logs can end up in a mess of messages that are only interpretable by the producer. Logs of inter-module messages nowadays are mostly encoded in XML. Messages are structured in a header, containing the source of the message, a message identifier, and timing information. The latter is extremely important and time should be synchronized over all modules. The contents of the body of a message is defined by the developers of the module that writes the message and must be parsed by all modules that read it. Loggings can become extremely large, making it very cumbersome and difficult to investigate failures in the communication protocols by hand. Today, no tools exist that support module developers who use muP as the integration platform in the process of debugging the distributed system messages. The tool μeval contains knowledge about the message content and is able to parse messages produced by all current COMIC modules. It is designed such that it can monitor any message log that contains:

```
header: <timestamp> <id> <source>
body: any xml-encoded string sequence
```

For example, if a user interacting with the system would draw a wall and speak out its length, the following message sequence would be recorded:

```
<msg>t0 id0 pen-tablet
some-sequence-of-coordinates</msg>
<msg>t1 id1 microphone
some-audio-input</msg>
<msg>t2 id2 PII
some-wall-encoding</msg>
<msg>t3 id3 ASR
some-lattice-containing-length</msg>
<msg>t4 id4 FUSION
some-wall-with-length-encoding</msg>
<msg>t5 id5 DAM
some-rendering-and-next-state</msg>
```

In this example, it is assumed that all input data are communicated, including audio signals. In most cases however, audio and video signals do not pass through communication channels in order to reduce bandwidth. This is also the case in COMIC, where the ASR system is directly coupled to a microphone and stores audio fragments on disk. Pen coordinates are communicated and are thus contained in the multimodal system logs.

Fast semi-automated annotation of MM interaction

When annotating multimodal interaction dialogs, the annotation process in general takes at least as long as the interaction itself. By using μeval, this process can be sped up considerably, while recording performance statistics for the individual modules. The tool considers header information present in the system logs, and sorts messages by their source and timestamp. So, messages from all PII, ASR, and other sources can be identified and categorized. For each message, messages from other sources that temporally correspond to it, can be detected. User input can be monitored by depicting pen input coordinates and playing audio inputs stored on disk. The latter is possible when ASR messages are marked up with the filename of the corresponding audio fragment. Now, during processing of the recorded loggings, for each sequence of messages, the user input is rendered and the corresponding output of each module is presented in a readable manner. The evaluator of the interaction dialogs can judge each output as being ok, false, rejected by the module, as noise, or out-of-grammar. All correct interpretations can directly be used as the label of the unknown user input, and require no further involvement of the evaluator. All other classes of input can be stored for later processing or can be transcribed manually. We have used μeval effectively for evaluating data from the current human factors experiments. Evaluating each experiment took about 15 minutes, whereas the original interaction took on the average 60 minutes.

The labeling and annotation scheme that we employed is a simplified version of more elaborated schemes that are currently applied in studies that specifically focus on unraveling the interaction between human and machine in considerable detail (e.g. Shin et al. 2002). Elaborate tagging schemes are used to study the performance of the individual recognition and processing components in large dialog systems. In one such a system (Kazemzadeh et al, 2003), tags were divided into three classes (see also http://sail.usc.edu/dialog/model_tags.html). The following classes can be distinguished.

- SYSTEM: explicit confirmation, implicit confirmation, help, system repeat, reject, non sequitur.
- USER: repeat, rephrase, contradict, change request, start over, scratch, ask, acquiesce, hang-up.
- TASK: error, back on track, success.

The four labels that we have applied for the analysis of the HF experiment log files (i.e., ok, false, noise, out-of-grammar) form a subset of this larger set, and were chosen to make a complete turn-by-turn annotation of all dialog steps feasible in reasonable time.

Data collection and labeling using μeval

All logged data have been processed using our evaluation tool. For each system prompt, the expected class of user response is known (i.e. wall, window, door, or some measure). For each individual module, a label was assigned by the human evaluator to indicate the correctness of the module output. Rejects or confirmations by the user or by each system component were also labeled accordingly. All data that were interpreted by a decoder and were labeled as 'ok' by the evaluator can be considered as a candidate for automatic transcription. Depending on the recognition performance of the decoding systems, this can speed up the transcription process considerably, as both segmentation and labeling are performed automatically. Cases where the system is unable to handle the input correctly are of special interest for improvements. Also data that are rejected by the recognizer, e.g., because the user draws an unknown shape, or in cases where the user employs out-of-context speech, are interesting. For speech, these data are used to refine the language model and to tune acoustic garbage models. For pen input, these cases form examples that require new pattern recognition algorithms.

The distinction between the decisions made by the DAM and the decisions overruled by the human wizard is not considered in the labeling process, as the focus of this research is on the input recognition and fusion technologies.

Users from WP3 and WP4 in different labs (DFKI, NICI) have used μeval for labeling and debugging purposes. It has proven to speed up both processes considerably.

Chapter 4

Results/System performance

In this chapter we present the results of the eventual experiment, in which 28 native Germans used the system to enter the blueprints of three bathrooms. Although the focus of the experiments was on understanding the behavior of the subjects and their appreciation of the system, we start this chapter with a detailed analysis of the performance of the system. Thus, section 4.1 gives a global overview of the system performance. Then, there are four sections that describe the performance of the four major input modules: ASR, PII, NLP and Fusion. The last section describes the analysis of the questionnaire, yielding information about what subjects think about the system and the experiments they contributed to. In this section, the correlation between recognition performance, task performance and subjective measures is discussed.

4.1 Global evaluation of multimodal input recognition modules

Below, a number of tables are given that summarize the performance of the individual modules. In the sections to follow, this information is discussed in more detail, while presenting examples of situations in which certain modules fail. In addition, we will give several examples of behaviors of subjects that were not anticipated, and consequently caused recognition problems. The results presented here are automatically generated based on the information obtained through μeval. Using the time stamps in the header of the logged messages, the time distance between two subsequent semantic expectations issued by the DAM is defined as the total turn time. Average turn time was computed for the five input concepts and for each of the three entered bathrooms.

Table 4.1: Average timings for different bathrooms, per concept and modality. For each bathroom, for each concept, the average time per turn, the time for recording pen inputs (second columns) and speech inputs (third columns) is shown.

concept	Bathroom 1			Bathroom 2			Bathroom 3		
	turn	PII	ASR	turn	PII	ASR	turn	PII	ASR
WALL	11.4	4.1	2.9	11.0	3.6	3.0	10.5	3.6	2.8
DOOR	13.3	3.4	3.0	11.9	3.5	3.0	12.0	3.6	3.0
WINDOW	11.8	3.7	3.3	11.4	3.9	3.0	10.6	3.5	2.9
SIZE	12.2	3.9	3.4	11.8	4.1	3.3	11.8	3.9	3.2
CONFIRM	12.9	2.8	2.9	12.7	3.0	3.2	12.6	2.9	3.3
overall	12.3	3.6	3.1	11.8	3.6	3.1	11.5	3.5	3.3

Prior to each session a calibration of the ASR input level is performed. This adds about 18 seconds to the total duration of a session. The number of turns does not change over bathrooms (respectively 34.8, 35.0, and 34.9 turns). The average turn time shows a tendency to decrease (12.3, 11.8, and 11.5 seconds) as users get more experienced. The difference between the second and third bathroom is not significant, which indicates that subjects quickly understood the task and that the instructions they received are sufficient. Still, many users show adaptive and learning behavior, as discussed in the sections below.

When considering recognition results per input category, the Tables 4.2, 4.3 and 4.4 presented below indicate whether users improve their pen and speech input over time. Since the semantic interpretation of ASR output depends on the entire recognized sentence, string error rates rather than word errors rates are reported. For sizes interpreted by PII, also string error rates (e.g., ”3 m”, ”217.5 cm”) are reported. In most cases, FUSION is able to merge the PII and ASR input channels quite well. The column marked USR shows the subjects' reaction to the system's recognition hypotheses. In experiments like those described here, subjects may accept wrong recognition hypotheses because they have given up attempting the system to understand them. By doing so, the interaction can proceed to the next step. In a small number of cases subjects rejected correct hypotheses, either by mistake, or because they made a mistake themselves (e.g. saying ”2 meter 50”, while they meant to say ”two meter sixty”). For example, in Table 4.2, in 43 out of 319 cases, the user falsely accepted a wrong system response, or falsely rejected a correct response.

Table 4.2: Bathroom 1. n = number of dialog acts; ok = number of correct recognitions; fa = number of false accepts; r = number of unclassified (remaining)inputs.

concept	PII				ASR				FUSION				USR			
	n	ok	fa	r	n	ok	fa	r	n	ok	fa	r	n	ok	fa	r
WALL	119	117	0	2	41	19	3	19	148	147	1	0	115	112	3	0
DOOR	68	47	7	14	45	16	11	18	91	90	1	0	46	44	2	0
WINDOW	34	28	0	6	40	22	7	11	62	62	0	0	28	28	0	0
SIZE	190	123	61	6	201	66	81	54	340	336	4	0	319	276	43	0
CONFIRM					111	82	14	15	98	98	0	0				

Table 4.3: Bathroom2. n = number of dialog acts; ok = number of correct recognitions; fa = number of recognition errors; r = number of unclassified inputs.

concept	PII				ASR				FUSION				USR			
	n	ok	fa	r	n	ok	fa	r	n	ok	fa	r	n	ok	fa	r
WALL	117	114	0	3	37	25	5	7	146	146	0	0	114	112	2	0
DOOR	50	40	0	10	33	18	6	9	70	69	1	0	38	35	3	0
WINDOW	39	34	0	5	34	22	2	10	59	58	1	0	33	33	0	0
SIZE	216	139	72	5	219	84	105	30	392	382	10	0	363	326	37	0
CONFIRM	3	3	0	0	133	102	10	21	118	118	0	0				

A quick glimpse at these tables shows a number of interesting observations. First, the vast majority of confirmations are input through speech, although subjects were told that system requests for confirmations could be answered by entering any pen input. Furthermore, it is evident that there is a relation between the number of errors and the total number of turns. For each recognition result that is rejected by the user, the system re-phrases the question and another turn is recorded. This explains the different number of inputs (n) in the tables. Speech utterances during input of the graphical concepts wall, door, and window contain supporting phrases like ”I'm going to draw a wall here” or ”Here comes a door”. In many cases, these utterances are not recognized correctly (because the language model does not include all possible expressions). In the section on the ASR we will give a more detailed account of the ASR performance in which this and similar effects are taken into account.

Recognition performance for pen input interpretation is excellent in case of the recognition of drawings. The errors are caused by rather complex drawings that PII was not designed for. In general, the performance of both PII and ASR increases during the course of the experiment. These global observations of changing recognition results over different bathrooms are more visible when considering Table 4.5.

Here, recognition performance in percentage correct is displayed. Performance is defined as the number of correctly recognized inputs divided by the total number of inputs. For PII, the overall performance increases

Table 4.4: Bathroom3. n = number of dialog acts; ok = number of correct recognitions; fa = number of recognition errors; r = number of unclassified inputs.

concept	PII				ASR				FUSION				USR			
	n	ok	fa	r	n	ok	fa	r	n	ok	fa	r	n	ok	fa	r
WALL	116	116	0	0	52	30	5	17	147	146	1	0	114	112	2	0
DOOR	61	46	4	11	28	18	8	2	84	82	2	0	49	42	7	0
WINDOW	39	34	0	5	28	20	2	6	59	58	1	0	32	32	0	0
SIZE	198	149	48	1	235	89	109	37	388	379	9	0	358	319	39	0
CONFIRM	2	2	0	0	124	105	14	5	121	121	0	0				

Table 4.5: Average percentage correct recognition of the modules per bathroom.

concept	Bathroom1				Bathroom2				Bathroom3			
	PII	ASR	FUS	USR	PII	ASR	FUS	USR	PII	ASR	FUS	USR
WALL	98.3	46.3	99.3	97.4	97.4	67.6	100.0	98.2	100.0	57.7	99.3	98.2
DOOR	69.1	35.6	98.9	95.7	80.0	54.5	98.6	92.1	75.4	64.3	97.6	85.7
WINDOW	82.4	55.0	100.0	100.0	87.2	64.7	98.3	100.0	87.2	71.4	98.3	100.0
SIZE	64.7	32.8	98.8	86.5	64.4	38.4	97.4	89.8	75.3	37.9	97.7	89.1
CONFIRM		73.9	100.0		100.0	76.7	100.0		100.0	84.7	100.0	
overall	76.6	46.8	99.2	90.6	77.6	55.0	98.5	92.3	83.4	56.1	98.4	91.3

with 6.8% over the three bathrooms entered. For ASR, with 9.3%. So, although the number of turns and average turn time do not strongly indicate learning effects, these results show that somehow users are able to adapt their behavior, leading to more efficient dialogs. In the next sections, we will delve more deeply into this matter.

4.2 System performance from the ASR perspective

In this section, the results of the Human Factor experiments will be discussed with a focus on automatic speech recognition (ASR). Based on the analysis of the experimental data presented in section 4.1 it is clear that the performance of the ASR module can hardly be considered as adequate. Here, we will analyze the performance of the ASR module in more detail, to identify the parameters that are most decisive for the observed performance. During the experiments, the ASR proved to be quite sensitive to overall loudness levels, to the way of speaking (e.g. with inter-word pauses, with emphasis, slow), and to the way subjects verbally express their intentions.

Annotations, ASR-eval

The annotations of the log files, performed with μ-eval, provided a useful tagging of all human-system dialogues. In order to evaluate the ASR module in more detail, a separate analysis, ASR-eval, has been carried out. This analysis was based on a verbatim transcription of all the user utterances produced during the human-system dialogs. Table 4.6 shows an extract from the log file that was a result of this ASR-eval analysis.

The table shows only a very small fragment of the available information. Each line in the table represents a full turn (i.e. a system prompt + corresponding user reply). The first column denotes the (encoded) identity of the subject and the particular session. The second column shows the message number as specified by the log file from the module communication system (μP). The third column indicates the expectation in the prompt by the system. The fourth column provides the annotation given to this particular turn; it indicates the relation between the speech input (as actually uttered by the user) and the output of the ASR module (the result as actually recognized). In this column, an ’S’ refers to non-speech noise or silence, while ‘o’, ‘g’ and

Table 4.6: Extract from a log file. For the explanation of the column labels, see text.

```
 SID MID   Exp C  Sdur  Tdur S E D W R A
----------------------------------------
./S1  83  Wall S  2.26 19.96 0 0 1 0 0 0
./S1 156  Size f  3.16 12.13 1 0 0 0 0 1
./S1 209  Wall S  2.26  9.57 0 0 1 0 0 1
./S1 287  Size g  2.71 10.46 0 0 0 0 1 0
./S1 326  Size f  4.06 10.90 1 0 0 0 0 1
./S1 382  Wall g  3.02 10.88 1 1 0 0 0 0
./S1 422  Size f  3.55  8.10 1 0 0 0 0 0
./S1 466  Wall S  1.51  9.07 0 1 0 0 0 0
./S1 287  Size g  2.71 10.46 0 0 0 0 1 0
./S1 326  Size f  4.06 10.90 1 0 0 0 0 1
./S1 382  Wall g  3.02 10.88 1 1 0 0 0 0
./S1 422  Size f  3.55  8.10 1 0 0 0 0 0
./S1 466  Wall S  1.51  9.07 0 1 0 0 0 0
./S1 503  Size S  2.14  6.39 0 1 0 0 1 0
./S1 543  Size g  3.77 12.37 1 0 0 1 0 0
./S1 633  Wall S  5.18 13.61 0 1 0 0 0 0
./S1 696  Size S  2.85  6.46 0 0 0 1 0 0
./S1 780  Wall g  2.09 13.89 1 1 0 0 0 0
./S1 820  Size f  5.59  6.57 1 0 0 0 0 0
./S1 888  Wall g  3.28 13.80 1 1 0 0 0 0
...etc.
```

‘f’ denote correct, out-of-grammar, and incorrect (false) ASR responses, respectively. The fourth and fifth column denote (in seconds) the period during which the microphone was open for listening and the elapse time of the entire turn until the next turn. The six following columns contain binary flags that provide more information about the complete user action during the particular turn. These flags denote whether speech has been used, whether there is an erase command by the user, or whether a sketch, handwriting, recognizer reject, or a fixation of the previous beautification is done, respectively. So the ordering is {speech, erase, sketch, handwriting, sysreject, fix}.

The fragment shown presents the start of a session, in which the system starts with asking to draw a wall. The third binary flag equals 1, meaning that the user responded with sketching the requested wall. The following system prompt asks for a size. This size, however, is incorrectly recognized by the ASR (which is indicated by ’f’ in the fourth column in the second line). In the third turn however, the user decides to draw a new wall in response to the third prompt (by sketching again), thereby implicitly confirming (and fixing) the system interpretation in the second turn. Not earlier than the sixth turn, i.e. the third wall, the user starts to use the erase facility (indicated by the second binary flag).
The sequence of these tags gives a complete picture of the human-machine interaction. From these sequences, one can deduce the behavior of the user and the way the user adapts to the system.

For the 28 subjects analyzed, the complete table now consists of 2856 turns (this is including 97 SNR measurement responses), grouped into 83 sessions which amounts to about 35 turns per session on average. The number of turns per session ranges from 18 to 65. The absolute minimum of turns per sessions is 18; this number is based on the shortest session possible according to the dialog flow (cf. section3.2. As indicated above, the total number of sessions in this analysis is 83; a few subjects did more than 3 sessions, while the loggings of two subjects were partially incomplete. All these sessions have been taken into account in this study, because for ASR we are primarily focused on user behavior in relation to ASR performance within each session, rather than learning effects across sessions.

Table 4.7 gives an overview of ASR performance as a function of expectation of the prompt. There are five expectation categories: Size, Response, Wall, Door shape and Window shape. ’Size’ refers to the prompts asking for a length, i.e. basically a number plus a unit. ’Confirm’ (which is a shorthand for ’yes-no response’) is a category related to the expectation of a yes/no answer from the user. The expectation ’Wall’ invites subjects to draw lines. ’Door shape’ and ’Window shape’ refer to the system prompts for position of the door and the position of windows, respectively.

Table 4.7: ASR performance as a function of expectation (for the five expectations categories mentioned). The annotation ”False”, abbreviated ’f’, relates to the string error rate of ASR.

Expectation	Size	Confirm	Wall	Door shape	Window shape	total
False (f)	258	33	17	23	3	331
Correct (o)	279	260	51	36	3	629
OOG (g)	129	32	42	21	7	231
Silence/noise (S)	690	53	462	207	125	1537
Total	1356	378	569	287	138	2728

The column ’Size’ shows the performance (in terms of string error rate) of the ASR when a size has to be interpreted. The default performance measure for ASR systems, viz. ’Word Error Rate’, would show a more positive picture. However, especially for ’Size’ a single word error in a string of words can only be repaired by re-entering the complete string. Examples of possible user utterances conveying size information are: (English) ”The length is 4 meters 40”, ”3 meters 20”, and ”2 meters and 45 centimeters”. The system can deal with partial responses from the user in one specific case: if the user specifies a length by saying e.g. ’three’, the next system prompt will ask for a measure, in this case ’meter’ or ’centimeter’, to complete the size information.

The large number of observations for noise/silence is basically due to the specific task for the user in this experiment: many prompts asks for pictorial information, and many other prompts can be replied to using handwriting. Most subjects did not speak when drawing. If they did, most of these utterances were out-of-grammar (OOG).

When the silent, out-of-grammar and noisy inputs are left out of consideration, the string error rate for ASR over all turns is approximately 65 percent. However, the performance of ASR related to size prompts is worse: about 50 percent. For size prompts, the performance further declines to 41 percent if also out-of-grammar utterances are taken into account, and to about 20 percent when also the noisy utterances are included. Noisy utterances are the inputs to the system containing noise (background speech, mouth smacks, head set noise). We conclude that the robustness against non-speech sounds has to be improved to enhance the functionality of the ASR module.

It is relevant to observe that the percentages in Table 4.7 do not directly reflect the functional performance of the ASR as a module in a multimodal system. As observed above, the ASR has a string performance of about 50% in the case of a ’Size’ prompt, if we look at clean grammatical inputs. However, not all errors contribute with the same weight to the performance of system level as experienced by the user. The majority of the recognition errors relate to predictable confusions, such as between ”ein-zwei”, ”zwei-drei”, ”sechzig-siebzig”. But specifically annoying are the confusions between ’ein’ and ’nein’, which sometimes lead to unjustified implicit confirmation of an incorrectly recognized system interpretation. If that happened, it almost invariably caused confusion and irritation at the subject’s side.

Table 4.8 shows how often subjects choose for the speech or for gestures as function of the expectation in the prompt. The column ”Erase by pen” refers to the erasing by the user of the previous system interpretation.

The table shows that in 62% of the wall-prompts, subjects reply with sketching, in 26% they decided to erase the previous system interpretation. In 19% of all cases, they (also) used the speech modality, e.g. to say ’nein’ as a spoken user reject. Subjects try the speech modality in 48% of all cases in which the prompt asks for a size. Handwriting is used in 39% of all these cases. The high percentage for the speech modality for Size (48%) is primarily based on the structure of the dialogue. Most subjects try a speech response,

Table 4.8: The table shows how often subjects choose for the speech or for gestures (erase, sketch/draw, handwriting) as function of the expectation conveyed by the system prompt.

Category	Speech	Erase by pen	Sketch (draw)	Handwriting	total
Wall	0.19	0.26	0.62	0.00	569
Size	0.48	0.10	0.00	0.39	1356
Response	0.85	0.32	0.00	0.00	378

before eventually switching to handwriting. It will be clear that these data cannot be fully interpreted without knowledge of the dialog flow. The data are actually a result of a combined effect, in which three effects play a role: the real modality preference by the user (which we study here), the structure of the dialog (which is kept constant), and the encouragement to keep trying to get things done in the speech modality before switching to gestures (which we attempt to keep constant during the entire experiment by providing the same information to each subject).

In Table 4.9 more information is provided about the distribution of the patterns in the responses by the subjects. Each line represents a specific turn type that is characterized by the same 6 binary flags (speech, pen-erase, sketch, handwriting, system reject, and fix), followed by two counts. The first count presents the total number of occurrences, the second number represents the numbers for the expectation ”Size” only. This category has been chosen as it is the expectation for which we can expect the most interesting distribution of speech and pen use. The table presents all turn patterns that occurred more than 100 times, ordered according to the total frequency (column Total). As can be seen from the ”Total” column, the most frequent turn is using speech modality (speech flag = 1), at the same time fixing the previous interpretation (fix flag = 1). The next two most frequent patterns are using pen gestures for sketching and handwriting, also fixing the previous system interpretation.

The number of occurrences of turn types in response to the prompts with expectation ’Size’ is given in the rightmost column. From the second and third lines it can be seen that sketching is never used to specify a size - which is evident - and that handwriting is only used to specify a size and for nothing else. The last two columns show that the expectation substantially influences the balance between modalities. Remarkable is the relatively low number of instances of a combination of speech and pen erase gestures after an expected size, compared to the general case (line 2 in the table). Probably this shift is not related to a fundamental change in behavior, but rather to the specific ordering of prompt types in the experiment: many of the size expectations are responses to the three consecutive prompts for information about window features.

The majority of prompts about size are replied to by using speech (first line) or pen (fourth line), thereby fixing the previous interpretation. From the table (line 7 and 10), it can be deduced that in the case where no fixing of a previous prompt takes place, the pen modality is more often used than speech modality. These data also show the dependency of the dialog structure on the modality choice; the experiment set-up is such that it seems that speech is favorite in the case where there is previous input that can be fixed. As noted above, this happens mostly in a cascade of questions associated with the three consecutive prompts asking for sizes of window features at the end of each session.

Language use

As part of the ASR-eval, all utterances have been transcribed on orthographic level. Also noise sounds and other non-speech sounds, such as lip smacks, laughter and breaths, have been indicated. Of all audio files, about 41 percent are silent or contain noise. A total of 219 audio files (i.e. about 8 percent) contain speech fragments that were not transcribed on orthographic level because they contained out-of-task or out-of-domain observations. (These will be transcribed in a later phase, because these utterances are still useful for tuning of acoustic garbage models in ASR.)

Table 4.10 shows the most frequent speech fragments (whole user replies) with their counts of occurrence. ’Ja’ and ’nein’ are well represented: these are by far the most frequent answers to the prompts about the

Table 4.9: For an explanation see the text.

Flags						Total	Size
1	0	0	0	0	1	571	354
1	1	0	0	0	0	312	55
0	0	1	0	0	1	300	0
0	0	0	1	0	1	278	278
0	0	1	0	0	0	260	0
0	0	0	0	1	0	217	107
0	0	0	1	0	0	206	206
0	0	0	0	0	0	199	40
0	1	0	0	0	0	146	53
1	0	0	0	0	0	158	157

existence of other doors and windows. On top of that, ’nein’ is frequently used to signal a user reject. The language model of the observed utterances appears to be very simple.

The list of less frequent utterances has a typically long tail. A total of 337 different utterances occur just once in the database.

Table 4.10: For an explanation see the text.

Total	Annotation
272	nein
262	[breath]
199	ja
147	[...]
48	[noise]
36	meter
36	drei meter
33	zwei meter
24	ein meter
15	vier meter
14	zwei meter fuenfzig
12	das stimmt nicht
12	centimeter

Examples of rejections as uttered by subjects are listed in the following table. Those indicated with an asterisk (*) were included in the language model.

```
*  nein
   nein zurueck
   nein noch mal zurueck
   nein nein die angabe war falsch
   nein keine weitere tuer
   nein ich will [...]
*  nein ich meine [...]
   nein halt stop
   nein gibt’s [...]
   nein falsch [breath]
   nein falsch
   nein es gibt keine weitere tuer
   nein es gibt kein weiteres
   nein das will ich loeschen
```

```
*  nein das war falsch
   nein das war alles falsch
   nein das stimmt nicht [...]
*  nein das ist falsch
   nein [...] nein
*  nein [noise]
   nein [...]
   nein [laugh]
   hmm letzte angabe stimmt nicht
```

User adaptation

We here devote a brief discussion to the analysis of user adaptation. By user adaptation we mean the change in the subject's behavior as an effect of learning (adapting) about how to behave to the system, based on the (perceived) performance characteristics of the system.

Since the system prompt for a size can be replied to in two modalities, we specifically investigated how users deal with an ASR error after the subject specified a length. The most interesting cases, and also the most difficult to analyze, are sequences of system prompts and user replies that start with a specific error. For example: what happens after an ASR error of a length specification occurs, followed by an ”erase” of the subject, after which the subject is again prompted to specify the length? Two possible sequences, one in which the pen is used, and one in which the user replies with speech, are provided below:

```
Example 1.
./S3  73  Size f  2.46 12.96 1 0 0 0 0 0
./S3 186  Wall f  3.36 13.45 0 1 0 0 0 0
./S3 239  Size S  2.46 10.34 0 0 0 1 0 0
etc.

Example 2.
./S3  73  Size f  2.46 12.96 1 0 0 0 0 0
./S3 186  Wall f  3.36 13.45 0 1 0 0 0 0
./S3 239  Size o  2.46 10.34 1 0 0 0 0 0
etc.
```

The particular sequence of turns (incorrect ASR result for Size prompt, followed by pen erase, followed by another Size prompt) happens 92 times in the annotated data. Of these cases, the subject chose the speech modality again 55 times (i.e. 60 percent), escaped to handwriting 15 times, and retried another pen-erase command 36 times (these numbers add up to more than 92 since speech and erase commands sometime coincide within one turn). In comparison, we found that, if the ASR recognition is correct, the probability that the next reply is given by speech is much larger than 60 percent, namely 79 percent. These percentages show a high probability to stick to the speech modality after a correct ASR recognition, and a tendency to escape to the gesture modality after ASR errors. Between subjects, however, substantial differences were observed. For example, a small number of subjects consistently attempt both modalities (by starting in the speech modality and persisting in using speech, also after a long sequence of faulty speech recognitions, and only use the gesture modality after a long sequence of attempts by speech).

Although the analysis of sequences of turns is the most interesting endeavor, the size of the data set very soon becomes a limiting factor for the significance of such an analysis. For example, in the entire database, it occurs only six times that a subject retries 3 times *or more* to provide a length after an incorrect ASR recognition. The subject's behavior will probably depend on a range of factors, such as the precise phrasing of the various possibilities during the experiment by the experimenter, and the amount of hinting at other solutions during the experiment.

After an incorrect system interpretation occurred, subjects showed a large variation when they used speech to reply to the system in the next turn. The usual behavior in such a case is a repeat, a rephrase, a reduction of speaking rate, over-articulation, and iteration, or a combination of these effects.

In the beginning of an experiment, most subjects seem inclined to just repeat the utterance or repeat it slower. Rephrasing is certainly not often used. Over sessions, the tendency to switch to the pen modality after an ASR error seems to increase, but the effect is, as we noted before, subject dependent. To investigate this formally, user behavior related to system responses can be monitored efficiently by using the output of μ-eval as below:

```
msgid expectation PII ASR FUS DAM USR
00834 WALL_LENGTH  -   f   o  drei R
00835 WALL_LENGTH  -   f   o  zwei R
00836 WALL_LENGTH  -   f   o  zwei R
00837 WALL_LENGTH  o   -   o  3 m  F
```

In this example, the user said ”Drei meter” and rejected the output of ASR three times. FUSION made no errors in passing on the interpreted inputs and only after the third try, the user switched to the pen modality, which was judged as ’ok’ by the evaluator, corresponding to the confirmation ’F’ (Fixed) by the user.

Conclusion

In conclusion, the annotation of the log files are indispensable for the analysis of the behavior of subjects. In general, we could establish the main effects reported in the literature on multimodal interaction. Subjects do not immediately change modality after a recognition error. On the contrary, a number of subjects are persistent in attempting to obtain a correct result by using speech. Moreover, subjects show a large variety in this aspect.

General patterns can be clearly observed, however. The analysis tools clearly show the preference for users to use modalities as a function of the semantic expectation available in the system prompt. On average, the use of the speech modality after Size prompts is facilitated by correct ASR recognitions (to 80 percent), inhibited by faulty recognition (to 60 percent).

The analysis of pattern *sequences*, i.e. of the behavior of users in the context of a short-term history, is necessary to get a better insight in adaptation. For the study of very complex interactions, however, the amount of data readily becomes insufficient.

For the improvement of the ASR module with respect to robustness, the experimental data and the analyses of the log files of the Human Factors experiments are critical. On the basis of these data, HTK will be improved in two directions: more powerful normalization features to enhance acoustic, robustness, and more powerful grammatical modeling, in order to be able to deal with out-of-grammar utterances.

4.3 System performance from the PII perspective

The research questions for pen input recognition are focused on the cases where the pen input interpretation software had problems. The acquired data have been analyzed to compute the recognition performance and pinpoint problematic samples. Furthermore, the analysis assessed the multimodal behavior of subjects in these cases. Both types of analyzes are described below.

4.3.1 Assessing recognition performance

The recognition performance over all users, for individual and combined sessions, and for the different input categories are given in Table 4.1. PII was able to recognize almost all walls (98% for the first, 97% for the second, 100% for the third bathroom). For doors, the recognition rate was 69%, 80%, 75% for the three sessions, while window recognition was 82%, 87%, and 87%. Only a few pen-based confirmations were recorded, although users were instructed that the pen could be used for that purpose. These performance numbers show no differences for the three bathrooms. However, the performance figures suggest that users were able to adapt their handwriting in the case of measures (65%, 64%, 75%). From research with interactive handwriting recognition systems, in particular with personal digital assistants (PDAs), this effect is well-known. Users can adapt their handwriting such that the system understands them better. We do not observe a similar increase in performance for the graphical objects. It remains to be investigated why this is so.

By considering the column labeled *fa* in the tables presented in Section 4.1, the cases in which the PII made faulty accepts can be determined. For walls and windows, no errors were encountered. In 6 out of 143 pen input samples, doors were accepted where the user generated something else. The amount of errors in case of measures was much higher: 147/497. In the remaining situations, pen input was rejected by the recognition system. The latter two cases contain the interesting samples for this study. As one of our goals is to improve the capabilities of the decoders, the samples that are falsely accepted (should be rejected) or that are falsely rejected (should have been accepted) will be considered here.

4.3.2 Rejected graphical objects

In order to enable robust processing and transparency, recognition modules must be able to reject samples in cases where the confidence in recognition is too low. This is explained in more detail in Section B.3.3 which can be found in Appendix B. In total, 68 PII rejects were encountered, as listed below:

Table 4.11: Eight classes of PII rejects. Associated with each of these are one or more threshold values based on which the decision to reject input is made.

class of reject	reason of reject	number
1	door not wide enough	19
2	window not wide enough	13
3	not enough samples	10
4	no door edges detected	9
5	wall not horizontal nor vertical	5
6	door opening unrecognizable	4
7	window beyond wall	4
8	door beyond wall	4

It is beyond the scope of this report to show all examples of these cases. Most of the classes of PII rejects are self-explanatory. The classes 1–4 typically occurred in cases where the user took too long to enter the information. According to the turn-taking protocol, the user should finish his/her drawing within five seconds. In many occasions, like in the category where doors and windows are not wide enough, users where able to draw one line, indicating one of the edges of the door or window, but did not manage to finish their drawing. Some typical examples are shown below, which are generated by the tool μeval.

In the examples given here, the feedback of the system was very limited. It merely said that the input could not be recognized, after which the user was prompted to re-enter the information. When repeated system rejects were encountered, the system would explain to the user how a door should be drawn. The vast majority of subjects was able to alter their first inputs based on this information, so that the system could understand them.

4.3.3 False rejects

The majority of cases in which the system rejected attempts to input windows, doors and walls were judged as correct. However, in some cases the parameters and algorithms used to decide for a reject, failed. In particular for a number of doors, the confidence limits based on which the system rejected shapes were just below the

Figure 4.1: Example of a situation where the PII is unable to determine the opening of a door. In this case, the user drew a 2D representation of a door, rather than the expected prototypical blueprint shape. The recognizer was able to determine the door edges (location of the door with respect to the wall).

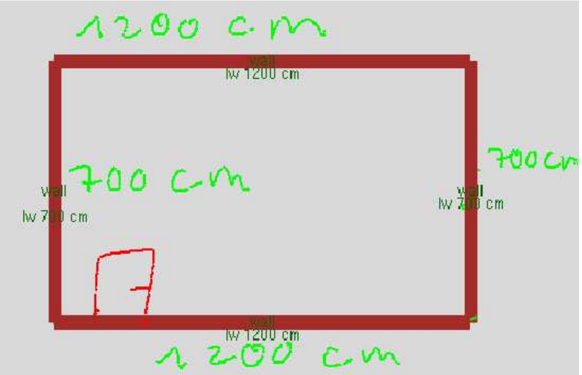
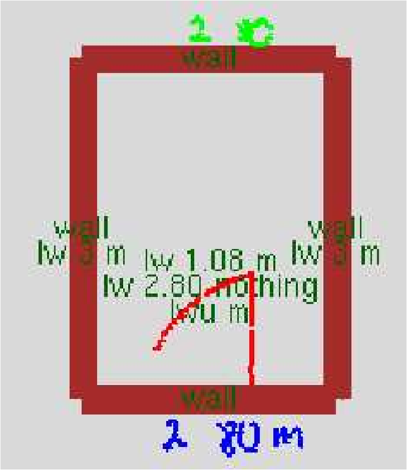


Figure 4.2: Example of a situation where the PII is unable to determine the edges of a door. The pen input displayed in red represents the user input that has been rejected. In this case, the curved line did not reach the wall close enough.



thresholds. As a consequence, PII research will focus on these cases to improve on the employed parameters and algorithms.

4.3.4 Errors in the recognition of measures

The reported recognition rates for digit strings with corresponding measures are relatively low relative to best results reported in the literature on pen input recognition. With respect to the false accepts (in total 181/604 errors were made), a number of typical failures of the recognition system can be identified. Most of the errors were caused by the very weak comma and dot recognition procedure. The system simply failed to recognize dots that were too large, or contained shapes that were similar to real digits. Furthermore, the digit string recognizer was not fit for recognizing commas in floating point numbers. The PII was trained mainly on input from a relatively small number of test persons, few of whom were native Germans, and who mostly used the dot to separate units and decimals. However, the data collected during the final experiment show that the comma is the default in German handwriting. As an example of failures caused by the dot recognition software, consider Figure 4.5.

Figure 4.3: Example of a situation where the PII is unable to determine the opening of a door. In this case, the user drew a shape that was not expected and that was too far off a wall. Furthermore, a typical digit confusion can be observed (208 *vs.* 200).

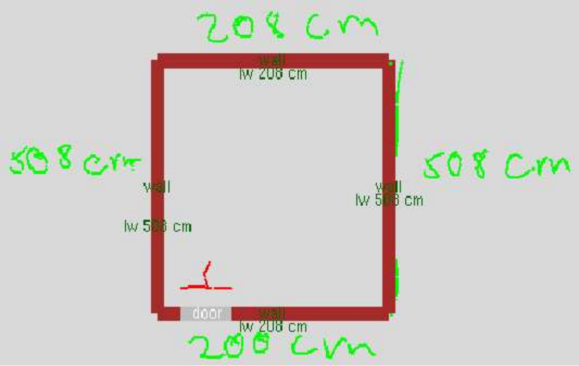
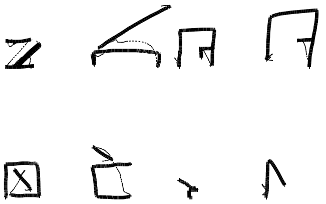


Figure 4.4: Examples of rejected doors. The system was not equipped with the recognition software to handle these cases.



Other errors were caused by unknown digits shapes. As mentioned before, the amount of German handwritten digits used for training the recognizer was limited. This caused in particular problems with the recognition of the German ’1’.

4.3.5 User behavior with respect to pen input

In most cases, users found the use of the pen as easy to use and experienced little problems in using it. The raw recognition performance of PII already indicates this: in on the average 80% of the cases, users could provide PII with input that was correctly recognized and accepted by the user. Also, users could handle the erasing functionality of the pen input recognizer without any trouble. Interestingly enough, the performance of PII increased over bathrooms with about 7%. This can only be attributed to learning and adaptation effects of the user. Based on the observed user behaviors, we distinguish three of such effects:

Figure 4.5: Some samples where dot recognition failed. The first comma and the comma in ’1,2’ was recognized as a ’1’. Most dots were produced by encircling a ’0’-like shape, resulting in the recognition of a ’0’, a ’6’ or an ’8’.

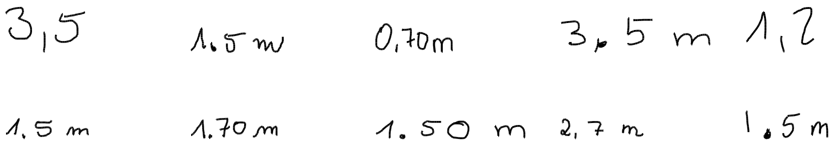
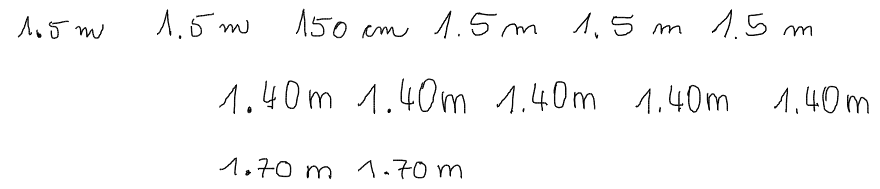


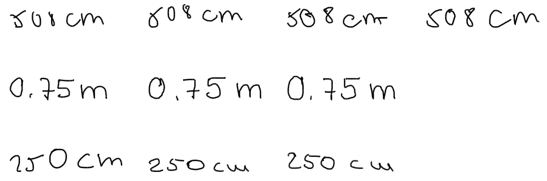
Figure 4.6: Typical examples of the German '1'. Users gave up after a number of tries.


1,5 m 1.5 m 150 cm 1.5 m 1,5 m 1.5 m
1.40m 1.40m 1.40m 1.40m 1.40m
1.70 m 1.70 m

- *persistence*, which means that users persist in entering information in the particular way that they want the system to recognize their input
- *changing*, which means that users change the way they generate information while trying to have the system understand what they mean
- *giving up*, indicating that after a number of tries, users just accept the wrong system response and continue with the next turn

These effects were particularly observable in the entering of measures. In some cases, like depicted in Figure 4.6, users just gave up. The system failed to recognize the handwriting, as the user kept producing a particular dot, or a particular digit shape. As an example of persistence, consider Figure 4.7.

Figure 4.7: Example of users persist in inputting measures using pen, without changing their handwriting.



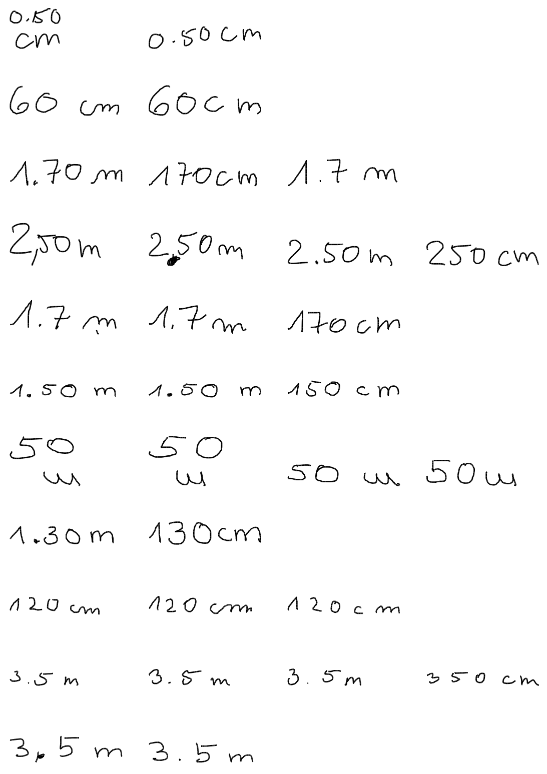
Here, users were able to make the PII correctly recognize their handwriting As examples of users adapting their handwriting changing, consider Figure 4.8 below.

4.3.6 Planned improvements based on these findings

Similar to the problems encountered by ASR, the pen input recognizer mainly failed in cases where unknown input repertoires were observed. These have to do with language-specific handwriting, a topic that is well-known from handwriting recognition. Although the performance of the PII was poor in these cases, on the other hand it is known that when users do their best in generating proper handwritten digit strings, this influences the signal in such a way that recognition will fail. This was anticipated during the design of the signal processing algorithms, which were focused on shape-based information in the signal rather than on velocity-based features. If this would not have been implemented, the examples presented in Figure 4.8 would all have failed.

Nevertheless, there are a number of important lessons to be learned from these outcomes. First, the pen input recognizer will be trained on the data acquired (and annotated) during these experiments. This will resolve problems with the recognition of German commas and typical digit shapes. Please note that these two problems mainly caused the relatively poor recognition rate of measures. Second, the parameters of the algorithms will be fine-tuned. In particular with respect to the reject criteria. And third, the data obtained through the first three bathrooms in the non-system driven condition will be explored, yielding valuable information on new graphical and handwritten shapes that need to be recognized.

Figure 4.8: Some examples of user adaptation.



These planned improvements will ensure a more robust processing of handwriting input. Furthermore, it is planned that the obtained experiences with how users accept system rejects will yield input for a new DAM. We have noticed in conversations with the subjects after performing the experiments, that they would have liked to see the reasons of reject. In the current set up, the DAM was not equipped to generate such feedback to the users. As user acceptance correlates with a transparent way of interaction, the system should be able to explain why a certain input is rejected. The pen input interpreter will be further developed with this observation in mind.

4.4 Results: Natural Language Processing

The human factor experiments have two impacts on the further development of NLP. First, the knowledge bases will be extended based on observed speech variants. Second, the experiments influence the further development of the built-in scoring function of NLP.

4.4.1 Extention of knowledge bases

During the experiments the subjects uttered several unexpected speech variants which were not covered by the ASR grammar and the NLP rule set and therefore, could not be processed. Some of the reasonable speech variants were already added to the knowledge bases, the rest of the reasonable variants will be integrated in the T24 demonstrator.

Examples for unexpected expressions for sizes are *eins punkt fünf meter (one point five meter), ein einhalb Meter (one and a half meter)*. Examples for unexpected feedback are *nein die Eingabe war falsch (no the* *input was wrong), falsch das letzte löschen (wrong, delete the last one), letzte Angabe korrigieren (correct last input)*.

NLP is also responsible for analyzing the PII interpretation of handwriting input. Therefore, hand writing variations were also considered. During the experiments only a few unexpected variants were detected like *5 m 30*.

Human factor experiments can also be used to increase the robustness of the rule set against speech recognition errors. Increased robustness means that although the utterance was only partially recognized a meaningful result is still produced. But since ASR uses a grammar as language model, the output of the speech recognizer is limited to this grammar and more predictable. Only a few rules were modified to increase robustness, e.g., by making some conditions optional. The situation will change as soon as n-grams are used as language model. Then the output of the recognizer varies much more and robustness of rules becomes more important.

4.4.2 Optimization of the scoring function

ASR provides as result not only a best hypothesis but an n-best list. An acoustic score is associated with each result. NLP uses an own scoring function which provides a score based on the semantic expectations provided by DAM, the portion of processed input, module internal semantic knowledge and the acoustic score. FUSION uses the score of NLP and internal knowledge to select the one hypothesis which is sent to DAM. An n-best list together with a smart scoring function within NLP can lead in principle to a better overall performance of the dialog system.

Using the collected data we wanted to find out to what extend the scoring function of NLP can improve the overall performance of the system and what a reasonable length for the n-best list is. Therefore, the following scoring functions were tested:

- the currently used built-in scoring function as reference
- a perfectly working scoring function which selects always the best results providing the upper limit for a real scoring function
- three modified scoring functions to get an impression of possible improvements

The length of the n-best list was varied for each test from $n = 1$ to $n = 10$ to determine the influence of the length of the n-best list on the results.

Since the lexicon of ASR will grow with an increased functionality of COMIC in the future and ASR will be further developed the presented results (and conclusions) are only preliminary and not definite. Therefore, the scoring function experiments will be repeated during the project duration. This can be easily done since the infrastructure exists now and the interfaces between the modules will be changed only in a backward compatible manner until the end of the project.

Setting

To perform the tests on the scoring function, for each turn the original audio data, the ASR output and the expectations provided by DAM were extracted from the logged data. Additionally, for each turn a gold standard for NLP output was provided based on the original audio data. Gold standard means that this is the output as it should be with a perfectly working NLP and ASR. The gold standard was produced by annotating the spoken utterances with the help of the recorded audio data and using NLP to produce the gold standard. NLP's output was checked afterwards manually. If the output was not correct the knowledge bases of NLP were adopted accordingly. This procedure is less error prone and faster than annotating directly the NLP's XML output structure. As a side effect the knowledge bases of NLP were improved at the same time.

Although several alternative results are sent to FUSION it is assumed that only the one with the highest NLP score is relevant. This is feasible since vast majority of verbal user inputs within the collected data is

Table 4.12: Number of correctly analyzed utterances using the default scoring function and varying the length of the n-best list.

n	all (1020)	size (374)	measure (65)	feedb. exp. (322)	feedb. unexp. (259)
1	671 (65,8%)	138 (36,9%)	50 (76,9%)	279 (86,6%)	204 (78,8%)
2	689 (67,5%)	140 (37,4%)	51 (78,5%)	291 (90,4%)	207 (79,9%)
3	699 (68,5%)	140 (37,4%)	52 (80,0%)	297 (92,2%)	210 (81,1%)
4	701 (68,7%)	141 (37,7%)	52 (80,0%)	301 (93,5%)	207 (79,9%)
5	701 (68,7%)	141 (37,7%)	53 (81,5%)	301 (93,5%)	206 (79,5%)
6	701 (68,7%)	141 (37,7%)	53 (81,5%)	302 (93,8%)	205 (79,2%)
7	701 (68,7%)	141 (37,7%)	53 (81,5%)	302 (93,8%)	205 (79,2%)
8	701 (68,7%)	141 (37,7%)	53 (81,5%)	302 (93,8%)	205 (79,2%)
9	700 (68,6%)	141 (37,7%)	53 (81,5%)	302 (93,8%)	204 (78,8%)
10	701 (68,7%)	141 (37,7%)	53 (81,5%)	302 (93,8%)	205 (79,2%)

unimodal, e.g., either speech or pen is used, but not both at the same time. So in these cases FUSION cannot use additional information to integrate verbal information from multiple sources. It can merely rely on a selection based on confidence values but it has to take over the selection of NLP.

The data collection contained 1197 utterances from the 28 subjects. Utterances which contain noise (84), which are aborted by the timeout of the ASR (35) or which are out of grammar (58) were not considered, so 1020 utterances remained. Since the recognition rate differed significantly whether the user utters a size, a measure or positive or negative feedback, a separate recognition rate was provided for each category. Feedback was additionally split whether feedback was expected by DAM or not. (In the other cases a splitting makes no sense since only one unexpected size and two unexpected measures were uttered.) Sizes were already considered as correct when only the value was correct but the measure was missing. This is justified as the value is the more important information and the measure can be determined either using context or by asking the user. If the value as well as the measure had to be identical the values in the size column of the tables would be 3%-6% lower.

Standard built-in scoring function

In table 4.12 the results for different n-best lists are shown using the standard built-in scoring function of the parser. The standard scoring function favors hypotheses that are in accordance to the expectation provided by DAM and next full utterances like feedback. A second less important criterion for the scoring function is the portion of input words used to generate the result structure. The acoustic score is the third criterion and only considered if the first two criteria provide no difference.

The table shows that the scoring function increases the overall recognition rate only by a maximum of +2.9%. The maximum is already achieved with $n = 4$. The single categories profit very different. Recognition of sizes is only improved little (+0.8%), but the recognition of measures (+4.6%) and expected feedback (+7.2%) increases a lot. As one can expect, the recognition of unexpected feedback drops slightly with a longer n-best list, since utterances with expected content appear on the n-best list. But the effect is reduced a lot since feedback is marked as full utterance in the ontology and rated higher than an utterance containing only a size or a measure that are marked as partial utterance. If size and measure are handled as full utterances, recognition of unexpected feedback drops to 67.8% (-11.4%) with $n = 10$.

The increased recognition rate of expected input with larger n-best lists is only the effect of the expectations provided by DAM. Turning off the second scoring criterion, the portion of process input does not effect the results.

Table 4.13: Number of correctly analyzed utterance using a perfect scoring function and varying the length of the n-best list.

n	all (1020)	size (374)	measure (65)	feedb. exp. (322)	feedb. unexp. (259)
1	671 (65,8%)	138 (36,9%)	50 (76,9%)	279 (86,6%)	204 (78,8%)
2	725 (71,1%)	163 (43,6%)	51 (78,5%)	292 (90,7%)	219 (84,6%)
3	751 (73,6%)	172 (46,0%)	52 (80,0%)	297 (92,2%)	230 (88,8%)
4	761 (74,6%)	177 (47,3%)	52 (80,0%)	302 (93,8%)	230 (88,8%)
5	764 (74,9%)	179 (47,9%)	53 (81,5%)	302 (93,8%)	230 (88,8%)
6	768 (75,3%)	181 (48,4%)	53 (81,5%)	303 (94,1%)	231 (89,2%)
7	769 (75,4%)	182 (48,7%)	53 (81,5%)	303 (94,1%)	231 (89,2%)
8	769 (75,4%)	182 (48,7%)	53 (81,5%)	303 (94,1%)	231 (89,2%)
9	769 (75,4%)	182 (48,7%)	53 (81,5%)	303 (94,1%)	231 (89,2%)
10	770 (75,5%)	182 (48,7%)	53 (81,5%)	303 (94,1%)	232 (89,6%)

Table 4.14: Number of correctly analyzed utterances when feedback is always regarded as expected.

n	all (1020)	size (374)	measure (65)	feedb. exp. (322)	feedb. unexp. (259)
1	671 (65,8%)	138 (36,9%)	50 (76,9%)	279 (86,6%)	204 (78,8%)
2	698 (68,4%)	139 (37,2%)	50 (76,9%)	291 (90,4%)	218 (84,2%)
3	712 (69,8%)	139 (37,2%)	50 (76,9%)	297 (92,2%)	226 (87,3%)
4	718 (70,4%)	140 (37,4%)	50 (76,9%)	301 (93,5%)	227 (87,6%)
5	719 (70,5%)	140 (37,4%)	51 (78,5%)	301 (93,5%)	227 (87,6%)
6	721 (70,7%)	140 (37,4%)	51 (78,5%)	302 (93,8%)	228 (88,0%)
7	721 (70,7%)	140 (37,4%)	51 (78,5%)	302 (93,8%)	228 (88,0%)
8	721 (70,7%)	140 (37,4%)	51 (78,5%)	302 (93,8%)	228 (88,0%)
9	721 (70,7%)	140 (37,4%)	51 (78,5%)	302 (93,8%)	228 (88,0%)
10	722 (70,8%)	140 (37,4%)	51 (78,5%)	302 (93,8%)	229 (88,4%)

A perfect scoring function

Table 4.13 shows the results for a perfect scoring function, i.e., a scoring function that magically choose always the right hypothesis within the n-best list. The perfect scoring function is simulated by checking if the gold standard result is among the produced results. The values show the maximum that is achievable by tuning the scoring function.

The table shows that a perfect scoring function provides an overall improvement of +9.7% (compared to +2.9% with the standard function). The recognition of size is improved by +11.8% (+0.8%), measures by +4.6% (+4.6%), expected feedback by +7.5% (+7.2%), unexpected feedback by +10.8% (+0.4%). This shows that a although a good scoring function can enhance the overall performance, no miracles can be expected.

Test of possible enhancements

Table 4.14 shows a first optimization of the scoring function. Since unexpected feedback happens quite often feedback is now always assumed as expected.

The overall recognition rate improves by remarkable +2.1% + compared to the standard scoring function. (Remember that the effect of original scoring function is only +2.9%). The recognition of unexpected feedback improves by +9.2% ($n = 10$), recognition of size remain almost constant (-0.3%), but recognition of measure drops by -3.0% ($n = 10$). This shows that the optimization depends on the relation between unexpected feedback and other input and may turn out as sub-optimal in the future.

Table 4.15: Experiments to improve the recognition of sizes. See text for details.

n	orig.	reas.	10%	30%	70%	100%
1	138 (36,9%)	138 (36,9%)	138 (36,9%)	138 (36,9%)	138 (36,9%)	138 (36,9%)
2	140 (37,4%)	147 (39,3%)	158 (42,2%)	156 (41,7%)	153 (40,9%)	146 (39,0%)
3	140 (37,4%)	148 (39,6%)	168 (44,9%)	160 (42,8%)	155 (41,4%)	147 (39,3%)
4	141 (37,7%)	146 (39,0%)	170 (45,5%)	163 (43,6%)	156 (41,7%)	148 (39,6%)
5	141 (37,7%)	147 (39,3%)	170 (45,5%)	165 (44,1%)	157 (42,0%)	148 (39,6%)
6	141 (37,7%)	147 (39,3%)	170 (45,5%)	165 (44,1%)	157 (42,0%)	148 (39,6%)
7	141 (37,7%)	148 (39,6%)	171 (45,7%)	166 (44,4%)	158 (42,2%)	148 (39,6%)
8	141 (37,7%)	148 (39,6%)	171 (45,7%)	166 (44,4%)	158 (42,2%)	148 (39,6%)
9	141 (37,7%)	148 (39,6%)	171 (45,7%)	166 (44,4%)	158 (42,2%)	148 (39,6%)
10	141 (37,7%)	147 (39,3%)	171 (45,7%)	166 (44,4%)	158 (42,2%)	148 (39,6%)

Table 4.15 shows some experiments to increase the recognition of sizes. First (second column) the values of sizes are restricted to reasonable values (20cm$<$size$<$5m) under the assumption that the subjects utter reasonable values. Since the language model of ASR is not restricted to reasonable values some false recognitions may be filtered out.

The second experiment checks if the recognition of sizes can be increased if the size can be approximated in advance. This is possible if the users has already drawn one wall and specified the size for it and if it is assumed that the user tries to draw further walls scaled. Since the evaluation of the drawing data is quite complex, we wanted to test first how great the gain is when sizes are preferred which are differ only by 10%, 30%,70% and 100% from the annotated value.

The restriction to reasonable values improves the recognition rate only slightly (+1.6%, $n = 10$). The prediction of the sizes improves the recognition rate much more (+6.7% with 30% precision) and the next step, checking the actual impact on the collected data, seems worthwhile.

Outcome for further development

The conclusions that can be drawn from the collected data are:

- The scoring function in combination with expectations can improve recognition rates but only in a moderate range (up to +10%).
- The right choice of expectations does not influence the overall performance dramatically, e.g., the added feedback expectation makes +2%. But within one category the effect is much stronger (e.g., +9%,-3%). Therefore, the expectations have to be carefully designed.
- Prediction of sizes can improve recognition rate (up to +6.7% with a precision of 30%).

Note that within this experiment the possible speech input is very restricted. Only sizes, measures and positive and negative feedback are possible. Therefore, the conclusions are only preliminary and must be repeated as soon as a wider range of input is possible. But as already mentioned this can be easily done since the annotated data can be reused and the infrastructure like the test programs is now available.

4.5 Results: Multimodal Interaction

In the following we will summarize the results of the human factor experiments with respect to their consequences for the FUSION component of the COMIC system.

Turn-Taking – Synchronizing the User and the System

During the development of the current system it turned out that the simplified turn-taking protocol that was used for the T12 system is not sufficient anymore as the possible multimodal interaction patterns evolved. So, we decided to implement a strict system driven turn-taking protocol that allows the user to express commands or requests within a fixed time frame (see chapter 2 for a detailed description of the turn-taking protocol). The system controls in any state of the dialog what will happen next – every system contribution ends in an implicit or explicit question to enter some information. This allows to predict quite reliable when the user is about to say something.

As a consequence to the limited time frame where the recognizers are listening, FUSION has to coordinate the different timeouts of the recognition/analysis modules based on the following schema:

- a message on the expectation pool sent by DAM signals that a system turn was finished and that now the user is supposed to speak or use the pen
- afterwards each recognition/analysis module send a recording started message, stating that it is now listing for input for a fixed period
- if no user input is detected during this period the analysis modules send a recording stopped message, stating no more messages will be sent until the next turn starts
- otherwise, if the user has provided some input via a particular modality the corresponding recognition/analysis module sends a recognition result

There are two different timeouts: (i) in case the user didn't provide any input the recognition module will wait for at least two seconds until it sends the recording stopped message and (ii) in case the user provided input the recognition modules will sent after seven seconds at the latest a recognition result. To inform users whether they can speak or use the pen there is a square depicted in the upper left corner of the screen. If the system is not listening this square is red and it turns to green when the system is listening.

To meet the requirements of the turn-taking protocol FUSION has to deal with three possible situations:

(i) both recognizers/analyzers sent a recording stopped message – so the user provided no input

(ii) one of the recognizers/analyzers sent a recording stopped messages after the first timeout but the other didn't – FUSION has to wait for the recognition/analyzing result

(ii) both recognizers/analyzers didn't send the recording stopped message and FUSION has to wait until the recognition/analysis is finished and then to integrate the results

During the human factors experiments it turned out that the subjects generally accepted this turn-taking protocol, however, in some cases utterances were truncated as they exceeded the second timeout. In nearly all cases the subjects tried to provide exactly those information the system requested. Only if several misinterpretations occurred in a row the subjects were confused what to say and when. Some subjects found it difficult to wait until the green square appeared on the screen. However, in most cases the turn-taking protocol was re-established after one system turn. Nevertheless, the fixed time interval where users are able to speak is to restrictive as it potentially truncates utterances.

As mentioned earlier the current dialogs are system driven and thus the user has not much freedom to enter other information than requested. The turn-taking protocol we implemented for the current system is very strict and will not be appropriate for mixed-initiative dialogs. Therefore, with respect to T36 we will need to extend the turn-taking protocol in a way so that both the user and the system can take the initiative. This would at least require to incorporate a second turn-taking protocol which doesn't restrict the time frame within which the user can do something.

Multimodal Interaction Patterns

As the current demonstrator does not allow for integrated multimodal input – it was out of scope to design a Wizard-of-Os DAM that would be able to process fused messages – we concentrate on multimodal effects like switching the modality, for example, if a previous command or request wasn't accurate processed by the system. An interesting effect that we observed is that subjects tend to switch the modality when a currently used modality lead to a series of recognition errors. However, although nearly all subjects showed this modality switch effect the extend how long this effect lasted varied heavily.

For example, many subjects first tried to enter size specification by speech, however, some of them were often misunderstood. So, they switched the modality and used the pen to enter size information. Important is here that some or most of the subjects at least tried to use the previous modality again, but some once they decided to switch the modality never switched back (see for example data-set subject21 or subject04).

An extreme amount of modality switches showed the subject subject12. While being asked to enter the size information for a wall she first tried four times to enter 2.80 meters by speech then she tried once with the pen, then switched back to speech, tried two more times and was finally understood when she switched back and used the pen. Interestingly she never tried to vary the style she entered the information. Some successful interactions later she was misunderstood when being asked to enter the size of a wall and she switched again multiple times the modalities.

Another interesting interaction pattern showed the subject subject00 in her first interaction in the system driven condition. When she was for the first time asked to enter the size of a bathroom she used both modalities to enter the information "5 meter". However, she didn't gave redundant information it rather seems that she recognized that the number ("5") she drew was a little illegible and said after she finished drawing *"five"* (the system treated this input as one multimodal utterance) but without repeating the measurement. The system misinterpreted this utterance and after having erased the size information she showed the same interaction pattern again. As we didn't anticipated this interaction pattern we will extend the FUSION component in order to be able to handle those in the T24 system.

As expected we could observe that all subjects show a strong preference to input graphical objects – like walls, windows, or doors – by pen. No clear preference regarding the modality could be observed for entering size information or measurements.

Some subjects showed a strong preference for pen input and used speech only for yes/no questions. Interestingly, those interaction patterns showed only such subjects whose handwriting was nearly perfectly recognized (consider for example subject13).

Outcomes for T24

The interactive process of developing a system proved to be very useful for the development of a robust and accurate module. Many bugs where found during the extensive system tests. However, uninformed users are the best system testers as they don't know the interaction patterns a module developer had in mind and so were other bugs discovered during the actual experiments.

In total FUSION produced during the experiments 29 erroneous outputs. However, as the errors of NLP and FUSION were considered together we need to go into more detail in order to analyze the errors made by FUSION. There is one bug which is responsible for nearly all errors made by FUSION. What happened was that under some circumstance the last expectation was output instead of the actual analysis result. This occurred for example eight times when FUSION was faced with objects drawn by the users (two times for walls, four times for doors, two times for windows). The other errors of FUSION where partly caused by NLP and the remaining errors also be traced back to this bug.

Based on the results of the experiments we revised the design of some production rules and we adjusted the initial weighting of the production rules. As a result we improved the overall performance of the FUSION component and in the current version of FUSION all above mentioned problems are fixed.

Additionally, we observed some effects in the recorded interactions that need to be addressed by the DAM for the T24 system. For example, it occurred quite often that a recognizer (PII or ASR) wrongly recognized a measurement of a size and this lead to implausible sizes of objects like a window of “100 m” length. Such a size for an window is obviously not possible, however, only the DAM can infer that since modules like ASR, PII, NLP or FUSION don’t have any information about the object a size correspond to. A possible way to resolve such a wrongly recognized size information is to ask a clarification question like *“Did you mean 100 cm?”*

In the same way the system should be able to identify when the interpretation of a size information must be wrong. For example, if the length of a window is greater than the length of the wall it is attached to or in case the user entered the size of a wall and this wall doesn’t fit the gap left between two other walls.

4.6 What subjects thought about the system

To analyze the subjective experience, subjects filled out a German version of a usability questionnaire that was adapted from an experiment on multimodal interaction carried out in an independent project [10]. This includes 26 questions about the user experience with and the behavior of the system on a five point Likert scale. The questionnaire is divided into three parts. In the first part, personal information is requested, about gender, age, education, handedness and creativeness. The latter question is not standard in questionnaires, but most relevant to the current set of human factors experiments. In the second part, detailed questions are asked about the computer experience of the subject. The amount of computer experience, drawing or writing with a tablet, experience with speech recognition systems, and programming experience is asked for. The third part, comprises 26 Likert scales. All scales were formulated in such a way that ’I was not able to form an opinion’ or ’I don’t know’ were not reasonable. Thus, neutral answers can safely be interpreted as ’I neither agree nor disagree’.

The results described in this section are based on the following list of questions. The replies from the users are summarized in Table 4.16.

4.6.1 Task understanding and rating of system feedback

Subjects reported no difficulties in understanding the task. The instruction was clear (mean 4.2) and most subjects understood what they were supposed to do while using the system (3.5). While the task was being performed, the system requested subjects to enter pieces of information. Upon recognizing the information, the system reacted by providing graphical and auditory feedback. Similarly, in case of errors, the system would provide relevant feedback to the subjects. The system prompts were rated as very clear (4.4). The rendering of `ascii` text and the beautification of drawings was also clear for the subjects (4.3). In cases where an input was wrongly recognized, most subjects knew what to do (3.7). These findings correspond to the subjective impressions of the experimenter. Subjects seemed to understand the task and the way the system provided them with information very well. However, some typical problems using the system could be observed:

- Subjects had difficulties waiting for sign of the system to proceed with answering a prompt. This was indicated by marking a square in the upper left of the drawing canvas with the color green. Some subjects tended to start immediately when a prompt had been given, almost as in human turn-taking behavior, in certain cases even before the prompt was ended. The result of this behavior was that (in particular for speech) subjects had to repeat their input. The small delay between the end of a prompt and the instance that the microphone or tablet was caused by the implementation of prompting. All system prompts were pre- recorded and played by engaging a certain program. Only upon termination of the program, tablet and microphone were opened after a small delay. The T24 demonstrator will feature a more advanced implementation, where the duration of each system utterance is computed in

Table 4.16: Mean, standard deviation and range of the scores on the Likert scales.

min	mean	max	sd	statement
2	3.6	5	1.1	Während der Arbeit mit dem System war immer höchste Konzentration erforderlich
2	3.5	5	1.1	Als ich mit dem System arbeitete war mir immer klar was ich tun konnte oder musste
1	2.8	4	1.1	Das System war einfach zu benutzen
2	4.2	5	0.9	Die Einfhrung am Anfang vor dem Experiment hat deutlich erklärt wie man Stift und Sprache in der Interaktion mit dem System verwenden kann
1	2.6	5	1.1	Ich würde das System gerne noch einmal benutzen
1	2.5	4	1.0	Ich fand das System effizient
2	4.0	5	1.1	Es hat lange gedauert, bevor ich alle Daten eingegeben hatte
1	2.4	5	1.2	Ich fand das System verwirrend
1	3.3	5	1.1	Ich fand den Gebrauch des Systems frustrierend
2	4.4	5	0.9	Ich habe die Fragen und Hinweise vom System immer gut verstanden
2	3.6	5	1.0	Die Kombination von Stift und Sprache war einfach
1	2.3	5	0.9	Ich hatte immer das Gefühl, dass ich das System unter Kontrolle hatte
1	3.2	5	1.1	Es fhlte sich natürlich an, Stift und Sprache zu kombinieren
1	2.4	5	1.4	Das System war für mich zu schnell
1	2.9	5	1.3	Während ich das System gebrauchte, fhlte ich mich angespannt
1	3.6	5	1.2	Ich fand den Umgang mit dem Stift einfach
1	2.1	5	1.1	Ich fand das System kompliziert
3	4.3	5	0.8	Ich fand es einfach, den Radiergummi zu benutzen
1	3.2	5	1.1	Die Benutzung des Systems hat mir Spass gemacht
1	3.2	5	1.1	Es hat oft zu lange gedauert, bis ich reagieren konnte
1	4.3	5	1.0	Die Art und Weise, mit der das System angab was es erkannt hatte, war eindeutig
1	2.2	4	0.7	Ich habe das System als zuverlässig erfahren
1	3.7	5	1.3	Wenn eine Eingabe falsch erkannt wurde, war mir klar, was ich tun musste
2	4.2	5	0.9	Meiner Ansicht nach muss das System noch stark verbessert werden
2	4.1	5	0.9	Ich fand es eine gute Idee, dass ich sowohl zeichnen und sprechen konnte
1	3.8	5	1.5	Das letzte Badezimmer war einfacher ein zu geben als das erste

advance, which allows for the system to be prepared to immediately open the input devices after this duration.

- It was difficult for some subjects to keep the pen at a sufficiently large distance from the tablet when the light was red. This resulted in some turns where the pen input could not be recognized even when the light was green. Although the system was designed to ignore pen input events when the tablet was closed, apparently the implementation was not operating as expected. However, this behavior of subjects ’hovering’ above the tablet is an interesting observation. Until this date, hovering information has not been considered in pen computing at all and may be used as an indication of a user planning to perform some action.

- Many subjects used commas instead of dots in written decimal numbers, at least in the first trials. This observation was already mentioned in the performance analysis of PII.

- Subjects had difficulties with the composition of measures. When only a number was recognized and it was impossible for the system to recognize the corresponding unit, subjects did not always understand that they had to erase the number first and then to enter the measure again. Some tried to enter the unit first and then to delete the whole measure, others tried to enter the whole measure again without deleting the previous number first.

- Some subjects had problems entering some specification during a five seconds interval. This was not explicitly stated in the instruction, but having failed to enter the information during this interval one time, a special prompt was given which made this clear. However, subjects did not always manage to be fast enough. On the other hand, subjects did not report that the system was too fast (2.4).

All previous findings suggest that the system should not be very confusing, which was also reported by subjects (2.4). Although subjects had no problems in understanding the task, they thought that the last bathroom was easier to specify than the first one (3.8). The latter effect sustains the findings from the previous sections, which show a tendency of improved system performance and reduced turn times over bathrooms.

4.6.2 Rating of the use of input modalities

When considering the use of speech or pen in the interaction, participants rated the pen as ”easy to use” (3.6), and the use of the rubber for erasing information on the screen as ”simple” (4.3). The combined use of pen and speech was rated as ”easy” (3.6). Subjects estimated the naturalness of this multimodal interaction as almost neutral (3.2), and they reported that they liked the idea of using both modalities simultaneously (4.1).

A correlation analysis of the ratings with ”easy of use” (2.8) and efficiency (2.5) of the system, showed a significant positive effect (r=0.5). Furthermore, subjects did not rate the system as complex (2.1) nor efficient (2.5). Subjects were neutral in rating the waiting time before being allowed to enter information (3.2). And overall, the need of system improvement was rated as high (4.2). The latter four ratings showed a significant negative effect with the ratings on the use of modalities.

From the experimenter’s impressions, it seems that subjects tended to use speech when possible (for measures & confirmations) most of the time in the beginning of the experiment, but probably due to the poor ASR performance the use of speech decreased after some trials.

Using a correlation analysis, it was found that the better the PII performance was relative to the ASR performance, the more the subjects evaluate the combination of pen and speech as easier. When looking at the performance rate for the PII and ASR for each subject, it was found that the PII performance was much better than the ASR performance on average. This explains why the more the pen was used, the more the system was rated as efficient, the more subjects got the impression that it took them less long to react, and the less they though the system needed improvement. Further, the better the PII performance, the stronger the subjects had the feeling of having the system under control, the less the system was judged as complicated, the more it was obvious what to do, and the less the system was rated as too fast. Finally, as may be expected, the ease of the pen usage was significantly correlated with the PII performance.

4.6.3 Rating of the system performance

In the previous sections we have reported on the recognition performance of the system in detail. The average performance for PII was 79.23% compared with an ASR performance of 52.6%. This relatively low performance is probably the main reason why the subjects are not very enthusiastic about the system. Subjects tended to describe that they needed to concentrate when using the system (3.6). They where neutral about the ease of use (2.8), stated that the system was not particularly efficient (2.5), and that it took long before all information was entered (4.0). There was a tendency for subjects to state that they could not control the system (2.3); also, the system was not considered as very reliable (2.2), and subjects though that the system definitely could need some improvement (4.2).

However, several subjects said that the system would have been easier to use if the recognition performance had been better. All in all, subjects would not like to use this system again (2.6), but almost every subject said that they would like to use an improved version of this system. Even if the previous findings did not pay many compliments to the system’s performance, the participants had fun using the system (3.2).

Again, using a correlation analysis, some significant results were found: A higher performance was positively correlated with less concentration, the system was seen as less complicated and less confusing, it was clearer what was recognized by the system, and the users felt less stressed during the interaction. It was apparent that the performance of the pen input recognizer determined most judgments of users. The performance of the speech recognizer had no significant effect on any question. A strong effect was found between the number of turns and the performance of pen input recognition. The less the performance, the more turns were needed.

Also, it was observed that the more the pen was used relative to speech, the more the system was rated as efficient, the less subjects thought that they had to wait for the system and the less they though the system needed improvements.

4.6.4 System performance dominates user ratings

Overall, there was a strong correlation between recognition performance and rated usability factors of the system. Interestingly enough, a strong correlation was observed between the number of turns and the willingness of users to use the system in the future. This seems contradictory, as in general a higher number of turns corresponds to more difficulties for the user to understand and complete the task. Apparently, as most users liked the application and the use of multimodal interaction with the system, they accepted the flaws and imperfections of the system.

Nevertheless, it can be concluded that the performance of the system dominates the user ratings. Significant positive effects were obtained between system performance and the factors described below, where the higher the system performance:

Table 4.17: User ratings with respect to system performance. It is apparent that system performance determines user acceptance.

required less concentration
thought the prompts were clearer (*)
were less fraught (*)
less thought the system was complicated (**)
thought that the system responses and feedback were much clearer (*)

were less confused (*)
thought the system was not too fast (*)
though the pen was easier to use (*)
took less time to react (*)
thought the system was easier to use (*)

(*) significant at 0.05 (2-tailed Pearson correlation)
(**) significant at 0.01

This summary and the results from the questionnaire described in this section indicate that recognition performance is crucial for the acceptance of the COMIC system for novice users. Although the system definitely needs improvements (4.2), it can be deduced from the questionnaire that this rating mostly refers to the recognition performance. Most users judged the way of interacting with the system using speech and pen as a good idea (4.1), the use of the pen as easy (3.6,4.3) and the feedback of the system as clear (4.3, 4.4).

Chapter 5

Summary and future directions

5.1 Conclusions

This document describes the construction of a Wizard-of-Oz system to conduct Human factor experiments with multimodal speech and pen input for a design task. In addition to the system itself, a tool christened μeval, was built to support researchers in processing the large amount of data that is generated during the experiments to extract objective data about the behavior of the subjects and the performance of the system and its individual components. Finally, the report presents the results of a large scale experiment in which 28 uninformed subjects entered the blueprints of three bathrooms of their own choice. After completing these tasks, the subjects completed a questionnaire comprising 26 Likert scales, addressing general and specific issues related to the task, the system and the interaction. A compilation of the objective measures extracted using μeval was used to predict the scores on the Likert scales. This provided insight in the relation between objective performance and subjective evaluations.

5.1.1 Building the system and the evaluation support tool

The architecture of the WoZ system was based on the T12 COMIC demonstrator system, in which all modules (except for the Wizard Support Tool) were successfully integrated. Despite the fact that we had functional and technical specifications of the WoZ system before we started to implement it, it appeared that neither the functional, nor the technical specifications were complete. As a consequence, the actual development of the system proceeded very much in the form of *rapid prototyping*. From our experience we think that it is fair to conclude that, given the state-of-the-art in the field of multimodal interaction, the implementation of almost every new application will require a number of iterations before the specifications of the total system and the individual modules are complete and consistent. It is essential that uninformed subjects are involved to test the performance and functionality of the system during the iterative development procedure. Failure to do so will result in a system that is extremely brittle, and that is likely to break down when used by anybody but the developers themselves. This will inevitably slow down the development of new multimodal applications, the more so if the team working on the implementation is large or when the members of the team are working in different geographical locations.

The specification and implementation of the tool (μeval) for analyzing the data logged during the experiments only started after we had a fairly precise idea of the information that we wanted to extract from the loggings. Therefore, the time needed to develop this tool was relatively short. Nevertheless, we are confident that this tool can be re-used with only minor changes for a wide range of experiments with multimodal interaction systems built with the *Multiplatform Testbed*.

5.1.2 User Interface

During the iterative development of the system much effort was spent on the design and the improvement of the user interface. Part of the problems we encountered are most probably due to the decision to use a strict system driven interaction style, an approach that has not been widely investigated in multimodal interaction, but that was forced upon us by the inevitable limitations of the pen input and speech recognizers. Yet, we are confident that most of the knowledge obtained in our experiment will generalize to more flexible interaction styles.

Subjects readily understood how to interact with the system using pen and speech for input. They also found it easy to understand the way in which the system used so called *beautified* graphical output to display its recognition results. We have seen few problems with the implicit confirmation strategy, which forced subjects to correct recognition errors immediately, because failure to do so would cause the system to consider the value as confirmed. To avoid endless repair loops, the interface did not offer the possibility to revert to previously entered data, so that corrections could not be delayed until after entering one or more additional data elements. However, subjects disliked the fact that it was impossible to correct only part of a complex item (such as a measure that consists of several digits and a unit, e.g. *3.5 m*) when only part of the elements were misrecognized.

To prevent loss of synchronization between complex multimodal inputs of the subjects and the interpretation of the individual elements of the input by the system, we were forced to define a strict turn taking protocol. Although this protocol, which effectively implements half-duplex communication, does not feel unnatural in the system driven interaction strategy, the implementation of the user interface appeared to be sub-optimal. The system showed a green square in the upper left corner of the pen input tablet as a signal for the subject that (s)he could start talking and/or writing in response to the system's prompt. Subjects found that the square was difficult to monitor because of its location at -or beyond- the periphery of their vision. In addition, the timing of the appearance of the trigger after the end of a prompt seemed to be less than fully predictable. Rather than spending more time on improving the timing and the visibility of the trigger signal, we will try to update the system architecture and the functionality of the individual modules so as to allow for a less rigid turn taking protocol, that should do away with the need for the trigger altogether.

5.1.3 Objective measurements

Not surprisingly, the performance of the two input recognizers turned out to be a major issue. Due to the lack of suitable training data neither recognizer could be trained appropriately. However, an ancillary goal of the Human Factors experiment was to collect application specific training data. This goal has been reached, so that we are now in a much better position to tune the recognizers.

The lack of application specific training data is a well known problem in the development of speech driven and multimodal interfaces. This problem will persist for a considerable time to come, but fortunately its importance will diminish over time as more building blocks in the form of grammars or language models for specific sub-tasks will become available.

By far the largest proportion of the recognition errors were observed in the size specifications for wall length, window width and window height. Unfortunately, many of the words and characters that are essential for the the specification of sizes are easyily confused. In future research we will adapt the user interface so as to avoid as many confusions as possible, for example by suggesting German speakers to say *zwo* instead of *zwei*, because the latter is easily confused with *drei*. In a similar vein, we will try to avoid *nein* as spoken negation, because it is easily confused with the numeral *ein*.

Recognition performance in our WoZ system has been negatively affected by the user interface problems alluded to in section 5.1.2. When users started to talk or write before the green square appeared on the screen, they almost invariably interrupted their input gestures, and tried to start over all again. However, these interruptions caused many disfluencies and time outs, which could not be properly handled or repaired by the input recognizers.

Recognition performance was also affected by the creativity of the subjects to invent expressions for pen and speech input that are quite natural, yet unexpected. The wide range of behaviors that we observed during the experiments will help us develop more powerful recognizers.

Due to lack of time and resources no attempt has been made to constrain the output of the pen and speech recognizer to values that are semantically and pragmatically plausible. For the same reason it has not been possible to implement the intelligence that is needed to prevent misrecognition of repeated attempts to input a size that boil down to the same -obviously incorrect- value. This facility is especially relevant because of the confusability of numerals, both spoken and written. The possibility to avoid repeated errors is extremely important for a system to be appreciated by its users.

5.1.4 Multimodal interaction

Due to the system driven interaction protocol, in which the subjects were prompted for information that lends itself either for graphical or for speech input, few simultaneous multimodal input actions were observed. Nevertheless, subjects appreciated the freedom to use the input mode that they thought would be most suitable. However, the lack of truly multimodal inputs made it very difficult for the Fusion module to prove its power and added value. Simultaneous use of pen and speech was most frequently observed in the form of spoken comments while a subject was drawing a wall or a door. However, these spoken comments were almost invariably out-of-grammar, so that the information in the speech could not be extracted. Not surprisingly, given the out-of-grammar status of most of these comments, few of them did contain relevant information.

The most interesting observation that we can make from our data is that subjects appear to prefer speech to input simple numerical data, such as the length of a wall. Only after repeated failures of the speech recognition systems do subjects revert to pen input. On the other hand, and quite naturally, subjects invariably use the pen when prompted to specify the shape of a room and the location of doors and windows.

We expect that the proportion of truly multimodal input turns will grow substantially if we can relax the strict system initiative. However, in order to make the most of the simultaneous pen and speech input we will have to design a user interface that unobtrusively constrains subjects to the set of graphical symbols that PII can handle, and to spoken expressions that are within the ASR grammar.

5.1.5 Likert scales

From the scores on the Likert scales it appeared that subjects understood the task, the system and the interaction style. Yet, they did not appreciate the system very much, mainly because of the large number of recognition errors and the inability to make partial corrections. From the analysis of the objective and subjective data independently and in their mutual relations it is evident that the performance of the input recognizers dominates the subjects' appreciation of the system.

5.2 Directions for future research

The conclusions drawn from the Human Factors experiment show the directions in which future research and development is most urgent:

- *Make the interaction protocol less rigid*
  For several reasons it is necessary, but also very challenging, to make the interaction between users and the system less rigidly system controlled. By relaxing the strict turn taking protocol and by allowing subjects to enter more complex information items in a single turn, the number of truly multimodal input gestures will increase. At the same time the number of disfluencies, mistakes and repairs is expected to decrease. Also, the contribution of Fusion to the performance of the overall system will become much more substantial. The challenges are for the input recognizers to handle the more complex input gestures.

- *Incremental processing and turn taking*
  The T24 demonstrator will still employ the current turn taking protocol. It is expected that research that

is under way in cooperation with WP2 will result in definitions of turn taking in multimodal interaction that come closer to the extremely flexible turn taking in human-human interaction. To support more flexible turn taking the input decoders must be capable of (probabilistic) incremental processing. This will allow for making predictions for the most probable semantic interpretation at any given time, which is paramount for better determining the end of the turn. Eventually, this research should result in a less constrained turn taking mechanism at T36.

- *Improvement of the robustness*
  The outcomes of the Human Factors experiments convincingly show that recognition performance must be improved. We will use the data collected in the Human Factors experiment to train better models for both PII and ASR.

- *Employing domain knowledge in the processing of user input*
  The current experiments clearly show that not employing common sense and domain knowledge may result in system responses that make no sense. In most existing systems domain knowledge and common sense reasoning are performed at the level of the Dialog and Action Manager, or alternatively in the Natural Language Processing and Fusion modules. In COMIC we will investigate the best possible division of work in this respect, where we intend to include the PII and ASR. After all, it is possible to constrain the language models in the input recognizers to prevent highly implausible output.

A particularly interesting issue for future research is how to use the semantic expectations that are generated by the Dialog and Action Manager. In particular when the dialog is less system driven and when the turn taking protocol is less strict, the input recognition modules will have to rely on techniques that can help to predict to which category the next user input will belong. The exploration of multimodal interactions with novice users in such a rich and unpredictable setting, will yield new insights in the required techniques. Moreover, it will provide new knowledge about the human adaptation strategies in task oriented multimodal interaction.

Bibliography

[1] T. W. Bickmore and J. Cassell. Relational agents: A model and implementation of building user trust. In *ACM CHI 2001*, Seattle, WA, 2001.

[2] A. Cooper. *About face: the essentials of user interface design*. John Wiley, 1995.

[3] G. Herzog, H. Kirchmann, P. Poller, *et al*. Multiplatform testbed: An integrat-ion platform for multimodal dialog systems. In *HLT-NAACL'03 Workshop Software Engineering and Architecture of Language Technology Systems (SEALTS)* (pp. 75–82), Edmonton, Canada, 2003.

[4] L. Almeida *et al*. User-friendly multimodal services - a MUST for UMTS. going the multimodal route: making and evaluating a multimodal tourist guide service. In *Proc. EUESCOM Summit*, 2001.

[5] J. Lee. *Words and pictures – Goodman revisited*. In R. Paton and I. Neilson (Eds.) *Visual Representations and Interpretations* (pp. 21–31). London: Springer-Verlag, 1999.

[6] J. Mankoff and G. Abowd. Error correction techniques for handwriting, speech and other ambiguous or error prone systems. Technical report, GVU, 2003. GVU Technical Report Number: GIT-GVU-99-18.

[7] S. Oviatt, P. Cohen, and *et al* L. Wu. *Designing the U-I for Multimodal Speech and Pen-based Gesture Applications: State-of-the-Art Systems and Future Research Directions*. In J. Carroll (Ed.) *HCI in the new millennium* (pp. 419–456), 2000.

[8] A. Potamianos, H. Kua, and *et al* A. Pargellis. Design principles and tools for multimodal dialog systems. In *ESCA Workshop, ISD-99* (pp. 22–24), 1999.

[9] S. Rossignol, L. ten Bosch, and *et al* L. Vuurpijl. Human-factors issues in multi-modal interaction in complex design tasks. In *HCI International* (pp. 79–80), Greece, 2003.

[10] J. Sturm and L. Boves. Effective error recovery strategies for multimodal form-filling applications. In *Speech Communication*, submitted.

[11] L. Vuurpijl, L. ten Bosch, and *et al* S. Rossignol. Evaluation of multimodal dialog systems. LREC 2004, in press, February 2004.

[12] W. Wahlster. Smartkom: Fusion and fission of speech, gestures, and facial expressions. In *Proc. First International Workshop on Man-Machine Symbiotic Systems*, (pp. 213–225), Kyoto, Japan, 2002.

Appendix A

Instruction and Questionnaire

Lies diese Anweisungen genau und sorgfältig durch und fülle das Formular aus. Folge den Anweisungen während des Experiments so genau wie möglich.

1. Erklärung

Vielen Dank für die Teilnahme an diesem Experiment. Es werden Audio- und Videoaufnahmen von dem Experiment gemacht, so daß später eine genaue Auswertung der Daten erfolgen kann. Alle Daten werden vertraulich behandelt. Bitte gebe hierfür Deine Zustimmung.

Lies das Folgende sorgfültig durch und unterschreibe an der angegebenen Stelle. Sollte etwas undeutlich sein, kannst Du dem Experimentleiter Fragen stellen.

Erklärung:
”Ich weiß, daß von dem Experiment Audio- und Videoaufnahmen gemacht werden. Ich gebe dem COMIC-Projekt hiermit meine Zustimmung, diese Aufnahmen zu Analysezwecken zu benutzen. Diese Aufnahmen werden nicht an dritte Personen außerhalb des COMIC-Projekts zur Verfügung gestellt. Diese Aufnahmen können allerdings für wissenschaftliche Prsentationen und Vorführungen zur Verfügung gestellt werden. Ich verzichte hiermit auf mein Recht die Videoaufzeichnungen anzusehen und zu beurteilen, bevor diese an andere COMIC-Mitarbeiter zur Verfügung gestellt werden.”

Name:

Unterschrift:

Datum: Nijmegen,

2. Hintergrund des Experiments

Dieses Experiment findet im Zusammenhang mit dem COMIC-Projekt statt. Für dieses Projekt arbeitet das NICI mit sechs weiteren Europäischen Partnern zusammen, um neue Interaktionsmöglichkeiten mit dem Computer - in diesem Fall Sprache, Handschrift und Stiftgesten - zu untersuchen. Das Projekt wird diese Interaktionsmöglichkeiten in ein bestehendes grafisches Programm zum Entwerfen von Badezimmern integrieren.

3. Erläuterungen zum Experiment

Das Experiment dauert ungefähr eine Stunde. Während des Experiments wirst Du einen Kopfhörer haben, worüber Du mit dem System sprechen kannst. Des weiteren kannst Du mit einem Stift auf einem digitalen Schreibtablett schreiben und zeichnen.

3.1. Anweisungen

Du mußt gleich zweimal drei verschiedene Grundrisse von Zimmern eingeben. Die Zimmer, welche Du eingeben sollst sind:
1) Dein Badezimmer,
2) Das Badezimmer Deiner Eltern,
3) Das Badezimmer Deiner Freundin/Deines Freundes,

Achtung: Du kannst nur rechteckige Badezimmer eingeben. Sollte eins der Zimmer eine andere Form haben, so mußt Du Dir ein rechteckiges Zimmer ausdenken. Jedes Zimmer muß außerdem mindestens ein Fenster besitzen. Sollte das nicht der Fall sein, so mußt Du dieses ausdenken.

Zwischen den zwei Teilen kannst Du eine Pause einlegen. Du kannst auch zwischendurch Pausen machen. Dies' mußt Du beim Experimentleiter angeben. Am Ende des Experiments bekommst Du eine Fragenliste, worin Du Deine Erfahrungen evaluieren und Anmerkungen machen kannst.

Teil 1

Danach kannst Du die drei Badezimmer nacheinander eingeben. Hierbei kannst Du den Stift auf dem Tablett sowie Sprache (über das Mikrofon) benutzen.

Außer den Grundrissen müssen noch einige weitere Informationen angegeben werden, wie zum Beispiel die Abmessungen. Du mußt für jedes Zimmer die folgenden Informationen eingeben:
1) Die Form vom Badezimmer. Diese besteht aus den Wänden und Ihren Abmessungen.
2) Die Position(en) der Tr(en) mit der öffnung.
3) Die Position(en) des/der Fenster, mit der Breite, der Höhe von der Unterkante und der Höhe.

Wenn Du mit einer Eingabe nicht zufrieden bist, kannst Du diese auch ändern. Du kannst die Rückseite des Stiftes als Radiergummi benutzen und vorherige Zeichnungen löschen. Bitte achte darauf, da Du nicht auf den Knopf auf dem Stift drückst.

Teil 2

Du wirst gleich ein Video sehen worin deutlich gemacht wird, welche Informationen Du von dem Zimmern eingeben mußt. Du kannst darin auch sehen, wie man den Stift und Sprache benutzen kann um Eingaben zu machen. Der Computer wird jeweils Aufforderungen geben, die angeben, welche Eingaben gemacht werden müssen.

Wichtig: *Während einer Aufforderung vom System, in der nach einer Information gefragt wird, ist es nicht möglich, eine Eingabe zu machen. Während dieser Zeit siehst Du ein rotes Quadrat in der linken oberen Ecke.* **Nur** *wenn dieses grün ist, können Eingaben gemacht werden.*

Wenn Du Eingaben mit dem Stift machst, achte noch auf folgende Punkte: - Wenn das Licht rot ist darf der Stift nicht das Tablett berühren oder sehr dicht darüber sein (weniger als 1 cm). Ansonsten mußt Du diesen kurz etwas weiter vom Tablett wegbewegen, bevor deine Eingaben erkannt werden können.
- Benutze nicht den Knopf auf dem Stift.
- Wenn Du Längen angibst, benutze einen Punkt anstelle eines Kommas für Dezimalzahlen.
- Bitte schreibe nicht kursiv und schreibe horizontal. Das Gitter wird Dir dabei behilflich sein.
- Ziffern in Zahlen (z.B. 1 0) und Buchstaben von Längeneinheiten (z.B. c m) dürfen nicht aneinander geschrieben werden.

Vor jedem Zimmer mußt Du ein wenig sprechen um den Spracherkenner zu kalibrieren.

Das Experiment beginnt, sobald Du bereit bist.

Questionnaire

Fragebogen Human Factors Experiment P-nummer:

Teil 1. Persönliche Informationen

1.1 Geschlecht: männlich / weiblich
1.2 Alter: . . . .
1.3 Händigkeit: links / rechts / beide
1.4 Höchster Schulabschlu: . . . .
1.5 Studienrichtung: . . . .
1.6 Hältst Du Dich für eine kreative Person:

gar nicht 1 2 3 4 5 sehr

Teil 2. Computer Erfahrung

Umkreise die Zahl, welche am besten Deine Erfahrung angibt mit:

2.1 Computer
sehr wenig 1 2 3 4 5 sehr viel

2.2 Zeichentablets mit Stift (z.B. Palmtop)
sehr wenig 1 2 3 4 5 sehr viel

2.3 Systeme mit Spracherkennung (z.B. Fahrplanauskunft der Eisenbahn)
sehr wenig 1 2 3 4 5 sehr viel

2.4 Programmierung von Computern
sehr wenig 1 2 3 4 5 sehr viel

Teil 3

Jetzt folgt eine Reihe von Behauptungen über das System, dass Du gerade getestet hast.

Bitte gebe deine Einschätzung auf den Skalen 1 (überhaupt nicht einverstanden) bis 5 (völlig einverstanden) an.

	Überhaupt nicht einver standen	Wenig einverstanden	Neutral	Einigermasen einverstanden	Völlig einverstanden
Während der Arbeit mit dem System war immer höchste Konzentration erforderlich Bemerkungen	1 ☹	2	3 😐	4	5 ☺
Als ich mit dem System arbeitete war mir immer klar was ich tun konnte oder mußte Bemerkungen	1 ☹	2	3 😐	4	5 ☺
Die Einführung am Anfang vor dem Experiment hat deutlich erklärt wie man Stift und Sprache in der Interaktion mit dem System verwenden kann Bemerkungen	1 ☹	2	3 😐	4	5 ☺
Das System war einfach zu benutzen Bemerkungen	1 ☹	2	3 😐	4	5 ☺
Ich würde das System gerne noch einmal benutzen Bemerkungen	1 ☹	2	3 😐	4	5 ☺
Ich fand das System effizient Bemerkungen	1 ☹	2	3 😐	4	5 ☺
Es hat lange gedauert, bevor ich alle Daten eingegeben hatte Bemerkungen	1 ☹	2	3 😐	4	5 ☺
Ich fand das System verwirrend Bemerkungen	1 ☹	2	3 😐	4	5 ☺
Ich fand den Gebrauch des Systems frustrierend Bemerkungen	1 ☹	2	3 😐	4	5 ☺

	Überhaupt nicht einver standen	Wenig einverstanden	Neutral	Einigermasen einverstanden	Völlig einverstanden
Ich habe die Fragen und Hinweise vom System immer gut verstanden Bemerkungen	1 ☹	2	3 😐	4	5 ☺
Die Kombination von Stift und Sprache war einfach Bemerkungen	1 ☹	2	3 😐	4	5 ☺
Ich hatte immer das Gefühl, dass ich das System unter Kontrolle hatte Bemerkungen	1 ☹	2	3 😐	4	5 ☺
Es fühlte sich natürlich an, Stift und Sprache zu kombinieren Bemerkungen	1 ☹	2	3 😐	4	5 ☺
Während ich das System gebrauchte, fühlte ich mich angespannt Bemerkungen	1 ☹	2	3 😐	4	5 ☺
Das System war für mich zu schnell Bemerkungen	1 ☹	2	3 😐	4	5 ☺
Ich fand den Umgang mit dem Stift einfach Bemerkungen	1 ☹	2	3 😐	4	5 ☺
Ich fand das System kompliziert Bemerkungen	1 ☹	2	3 😐	4	5 ☺
Ich fand es einfach, den Radiergummi zu benutzen Bemerkungen	1 ☹	2	3 😐	4	5 ☺
Die Benutzung des Systems hat mir Spaß gemacht Bemerkungen	1 ☹	2	3 😐	4	5 ☺
Es hat oft zu lange gedauert, bis ich reagieren konnte Bemerkungen	1 ☹	2	3 😐	4	5 ☺

	Überhaupt nicht einver standen	Wenig einverstanden	Neutral	Einigermasen einverstanden	Völlig einverstanden
Die Art und Weise, mit der das System angab was es erkannt hatte, war eindeutig Bemerkungen	1 ☹	2	3 😐	4	5 ☺
Ich habe das System als zuverlässig erfahren Bemerkungen	1 ☹	2	3 😐	4	5 ☺
Wenn eine Eingabe falsch erkannt wurde, war mir klar, was ich tun mußte Bemerkungen	1 ☹	2	3 😐	4	5 ☺
Meiner Ansicht nach muß das System noch stark verbessert werden Bemerkungen	1 ☹	2	3 😐	4	5 ☺
Ich fand es eine gute Idee, dass ich sowohl zeichnen und sprechen konnte Bemerkungen	1 ☹	2	3 😐	4	5 ☺
Das letzte Badezimmer war einfacher ein zu geben als das erste Bemerkungen	1 ☹	2	3 😐	4	5 ☺

— ENDE DES FRAGEBOGENS —

Appendix B

Technical research issues for individual modules

The current set of experiments has advanced the technology developed for the COMIC T12 demonstrator. In the T12 demonstrator, we successfully demonstrated that all COMIC components were functioning properly, integrated through the COMIC MULTIPLATFORM architecture. This chapter discusses the technological advances that have been made as a spin off from developing the HF research platform. Each of the modules used for the experiments is discussed: automated speech recognition (ASR), pen input interpretation (PII), the user-interface emulating the eventual Visoft tablet interface, natural language processing (NLP) and fusion. Furthermore, the Wizard support tool is presented, via which a human wizard can supervise the experiments.

Issues considered during these interactive experiments are robustness, correctness, efficiency and recognition performance. Based on the current functionalities, a number of detailed research questions can be formulated for each of the modules ASR, PII, NLP and FUSION.

B.1 Automatic speech recognition

As was the case for the T12 demonstrator, the automatic speech recognition is based on open source software for speech recognition purposes, the Hidden Markov Toolkit (HTK). HTK was originally developed at the Speech Vision and Robotics Group of the Cambridge University Engineering Department (CUED) where it has been used to build CUED's internal large vocabulary speech recognition systems. The current ASR version that we use is version 3.1 of the public domain software platform. HTK just provides a code base, not a recogniser: All the speech files for training of acoustic and language models, and files for testing have to be delivered by the user. Also the tuning of the recogniser, i.e. the adaptation to various test conditions (that may be different from the training conditions) is to be done by the user on the basis of appropriate tuning data. The original public HTK version allows recognition from file (disk) and from microphone (in two versions: end-of speech is controlled by keyboard or by automatic end pointing, while start-of-speech is always keyboard-controlled). In order for HTK to better serve a complex man-machine interface using a multi-module communication platform, the public HTK code base has been adapted to accommodate the constraints posed by the interacting modules. The most important issues concerning the ASR itself are manifest on three levels. First, the ASR requires a number of files in order to recognize incoming speech signals. These files (lexicon, acoustic models, language model, configuration files) are derived from corpora. For the present experiment, application specific corpora that are needed for development and evaluation of the ASR system were not available. Therefore the tuning of the ASR algorithms is based on reasonable, knowledge-based estimates of the various parameter settings. Secondly, considerable changes in the HTK decoder (HVite) were necessary to enable this software to operate one module in a multi-modular integrated system. The original code base is fully focused on stand alone off-line use. Thirdly, the ASR module is embedded in a so-called *wrapper* that takes care of the communication between the HTK speech recognition algorithm and the other modules.

The following list presents an overview of the ASR issues that will be addressed on the basis of the loggings of the human factor experiments described here.

- appropriate expansion of lexicon and language model. During the experiments, we have observed that the variety of expressions that subjects choose to convey information that is intuitive to them is large.
- start-of-speech and end-of-speech handling. This issue relates to the opening and closing times of the microphone. In the current set-up, the microphone is opened by the ASR after a command from the DAM to do so. This explicitly means that the speech from the subjects that is spoken before that moment is not captured for further processing and is thus lost. If this protocol works well, this means an advantage for the recogniser, since the problem of start-of-speech detection does not need to be solved. However, when something goes wrong in this protocol, for example because a subject starts speaking directly after the prompt, the recogniser looses track of the utterance and is therefore likely to provide incorrect answers, which leads to error-prone correction loops in the dialog.
- balance between word error rate and tuning of language model. The tuning of the language model is known to be extremely relevant for an on-line recognition task, and all logged data from the experiment will be used for this purpose.
- robustness. Based on the experience in this experiment, one concludes that the speech recognition module is not yet robust enough for on-line speech processing. To improve the robustness, HTK will be enhanced with channel normalization software. The logged data will be used as test bed for the associated experiments.
- out-of-grammar. An important issue is formed by the out-of-grammar utterances by subjects. This is related to a complex interaction between various human factors. The set-up of the experiment included a small square in the top left position of the tablet screen, its colour indicating whether the tablet was open or closed for input from the pen. The opening of the tablet was typically one second after the end of the audio prompt. Some subjects first responded to the audio prompt, and repeated their response after the square on the tablet changed colour. Although intuitively entirely correct, the resulting user reply is likely to be out-of-grammar and therefore inadequately recognized. In future research we will approach this problem along two independent lines: improving the user interface to reduce the number of disfluencies and hesitations, and improving the ASR system to be better able to deal with out-of-grammar utterances.

In addition, two issues will become of special interest with respect to the functioning of ASR in the context of a multimodal interactive system:

- the alignment between words, especially deictic words, and pen gestures
- the use of meta-linguistic information, such as prosody, in the utterances by the subjects, to detect their ’commitment’ with the linguistic information conveyed by the utterance

Although the performance of the ASR is still quite poor, the module itself was appropriate to support the scientific aims of the experiments. To enhance the appropriateness of the speech recognition module, the test lexicon and language model will be adapted to-wards the foreground (application) task on the basis of the logged data. The lexicon will also be expanded in two other ways: firstly by adding garbage models to capture words that are not in the lexicon, and secondly by adding words in order to cope with corrections by the user. In the first phase of the scheduled HF experiments, these corrections are only of a simple grammatical structure. For the final COMIC system, the eventually expanded lexicon will contain a set of task-related semantic keywords (about 300-400 words), including words to correct and reinterpret the

outcome of the recogniser. This extension allows the user to flexibly use the speech mode, including alternation and integrated use.

Fragile error handling currently remains the single most important interface problem for recognition based technologies such as for speech and pen input. Multimodal interface designs that combine the two input modes tend to have superior error handling characteristics (e.g., by Oviatt's group and many other groups). In order to perform experiments that go beyond the state of the art, ASR must support the various levels of disambiguation across modes, support dialog processing techniques on error avoidance and resolution. In the current human factor experiment, these issues are in principle addressed by cross-modal disambiguation and cross-modal error resolution).

B.1.1 Detailed research questions for ASR

The following research questions are important for the development of the ASR module. Some of them have already been investigated on the basis of the

annotations of the logged data, insofar they became available during the experiments.

- Adaptive behaviour of subjects and associated appropriate adaptation of the lexicon and the language model. Subjects apply a large variety of expressions to convey information that is intuitive to them. Especially the variety that subjects use after ASR recognition errors is interesting.
- start-of-speech and end-of-speech handling. The aim in T36 is to relax the constraints from a system-driven condition to wards a more mixed-initiative dialog. The way to avoid the recogniser loosing track of the utterance will be studied on the basis of the logged data.
- The balance between word error rate and tuning of language model will be one of the research topics. The tuning of the language model is relevant for an on-line recognition task, and all logged data from the experiment will be used for this purpose.
- robustness. The experiments with real-life on-line audio recordings have shown that the ASR module shows a lack of robustness and of channel normalization capability. To improve this robustness, HTK will be enhanced with channel normalization software. The type of normalization will follow the start-of-the-art in this domain. The logged experimental data will be used as test bed for the associated experiments.
- out-of-grammar. As already explained, a number of factors invited subjects to utter ungrammatical sentences, or utterances with a long pause between words. These utterances are likely to be out-of-grammar and therefore inadequately recognized. The adjustment of the language model without introducing new decoding problems is necessary for the improvement of the functionality of the ASR module.
- time out settings. In the current implementation, a 'wait time out' of 2 seconds has been implemented. That means that, from the moment the audio prompt is finished, subjects have two seconds time to start talking. Once talking, they have a 'speech time out' of 5 seconds to utter what they want to say. For many subjects, the 'wait time out' appeared too short.
- paralinguistic information. The logged data will be used to investigate to what extent pitch and durational cues can be used to indicate a measure of 'commitment' of the user with the linguistic content of the utterance. Beyond any doubt, a measure of irritation would be a very useful indication for the extent to which a subject is becoming irritated by sequences of incorrect recognition results. Currently, the importance of this type of research is acknowledged by an increasing number of speech and dialog research groups.

B.2 The user-interface

The user-interface (UI) is based on the SLOT software, which was used for the multimodal interaction experiments performed in WP2. The UI comprises a rectangular drawing canvas, on which the user can generate "ink" by use of the pen. Any ink produced by the user is rendered directly on the screen, in the colour red. The user is able to erase previous inputs, by using the back of the pen. Traces of ink that are recognized by the system, is indicated by the colour blue. This colour indicates to the user which pieces of ink have been used by the system for the generation of the recognition result. Recognition results is rendered (beautified) as described in the technical specifications and as discussed with Visoft (not included in this document). The user is able to correct misinterpretations made by the system by using the eraser functionality of the pen, operating on the blue ink and last beautified objects only. The system does not explicitly ask for a confirmation of the interpreted information. It asks for new information according to the dialog specifications, which are specified in the afore mentioned technical specification. If the user generates new ink or utters a non-rejecting speech act (implicitly starting to answer the new questions), the system will assume that the user judged the previously interpreted information as correct, and render the blue ink as green, thereby fixating this information. The UI mimics the eventual "service application ViSoft" (SAV) module.

B.3 Pen-input interpretation

The COMIC T12 demonstrator contained the first implementation of 2D gesture recognition. Although limited in functionality, this result is quite important, as it provides the infrastructure for building more advanced recognition components. The T12-recognizer was able to recognize deictic gestures (tapping and encircling). The pen input recognition system used for the HF-experiments comprises different recognisers for the modes that we expected the subjects to use: graphical (sketching) gestures, digits, and characters. Based on the system-driven dialog of the HF experiments, the distinction between these modes is made by considering the semantic expectations evoked by the DAM. In this section, the required (new) software developed for the HF demonstrator is described.

B.3.1 Graphical gesture recognition

The preliminary inventories made in pilot experiments indicate the basic graphics categories that have to be distinguished: (parts of) walls, (parts of) doors, (parts of) windows, and deictic gestures (pointing, encircling). Below, the way in which the different categories are recognized and rendered, is described.

Door, window and wall recognition

The current recognition system exploits the geometric (or structural) pattern recognition paradigm. In this technique, context-dependent knowledge about how humans produce the different categories is of paramount importance. For example, consider the surrounding walls of a bathroom, comprising of a set of individual walls (in the current experiments: four). Many people produce one rectangular shape, when they are allowed to draw the surrounding walls in an unconstrained fashion. Other people use a sequential ordering (left-wall; top-wall; right-wall; bottom-wall), although we have also observed less constrained sequences. An often accompanying problem with this structural approach is that it suffers from between-subject and even within-subject variation. This is a general problem in pattern recognition, that however should not introduce many problems for this task. Given that the basic building block (a line representing a wall) can be recognized accurately, the number of different configurations is limited. Furthermore, as the current experiments employ a system-driven dialog, we do not expect recognition problems in the case that lines have to be recognized as walls. Still, the outcomes of the experiments should validate this assumption.

When considering the way doors and windows are rendered in professional blueprints, only a limited number of different shapes exist. However, novice users of the COMIC system are in many cases not aware of how such renderings should look like. Therefore, it may be expected that subjects will be biased by the instructions. Although the exploration of the data acquired through previous experiments reveals that the number of ways in which a subject can draw a door or window is rather large, the gesture recogniser should not have serious problems with these objects. Recognition should further be enhanced by the fact that the current system-driven dialog prompts for these three required pieces of information on a step-by-step basis.

Pointing and encircling recognition

We have sustained the observations in literature [7], that when users use speech and pen in design applications, in many occasions the pen is used to indicate which object on the screen is being referred to. The “encircling” recognition software from the T12 demonstrator can be used for recognizing areas marked on the screen. The pen input recognition module maintains an internal representation of the current screen state to determine whether such encircling enclose (refer to) a wall, door, window or some text. For the recognition of pointing gestures, a new recognition module was designed. The recognition module is based on the acquired data collected in COMIC so far and a set of existing training data, provided by Daniel Fonseca. Encircling gestures and pointing gestures will not be beautified by the system.

B.3.2 Handwriting recognition

NICI has ample experience in handwriting recognition. Most of the classifiers used for recognizing handwritten text and digits are trained on a publicly available dataset, called UNIPEN. Although this data contains handwritten information of thousands of writers, the amount of German writers is very limited. About 200 German words (of city names) are encoded in the recognition systems employed by NICI. This introduces one of the challenging technological research issues we are faced with in COMIC: We will need to develop writer-independent handwriting recognition components, without having large amounts of training data available for the domain of bathroom design and without an extensive shape lexicon of handwriting building blocks produced by German writers. Therefore, the current set of experiments is considered as a first step in the bootstrapping process of generating increasingly more advanced recognition components. The data acquired in the HF experiments will be used for further training the current handwriting recognisers. For handwritten user input, we distinguish three input modes: (i) digits, (ii) isolated characters, and (iii) words. We will not pursue sentence recognition in COMIC. Based on the previous sets of experiments, we can justify this decision, as subjects do not produce complete sentences in bathroom design applications. For the current HF experiments, word recognition is not implemented as well. The instructions say that digits and characters must be entered in an isolated fashion. If subjects write their input as isolated characters, the need for cursive word recognisers and segmentation into characters is less high.

B.3.3 Detailed research questions for pen input interpretation

For PII, a number of design considerations were taking into account in order to ensure robust processing and transparent system rejects.

Robust processing of pen input requires a low error rate. Even if the recognition performance is not perfect, users accept rejects much more than false accepts, in particular in cases where the system is able to explain why a certain input is rejected. Each of the four mode-specific pen input interpretation systems is trained and tested on data acquired in the pilot experiments and in a series of uni-modal interactive experiments with expert users from our lab. Furthermore, the digit recogniser is trained on an elaborate data set containing about 60.000 isolated digits. The lower case character recogniser on more than 200.000 isolated characters. In the uni-modal conditions, tests with these systems yielded error rates of respectively less than 1% and 2%. For interactive tests with expert users, pen input interpretation appeared accurate as well.

However, first tests with novice users revealed that there were cases where the system persisted in yielding wrong results. As users found this unacceptable, each of the recognition systems was adapted such that it was able to reject user input that did not reach a sufficient confidence. For example, if a wall was requested and the user drew a line too diagonal, this resulted in a system reject adorned with the reason of rejection.

Likewise, in cases where a window shape was entered that was not in the familiar shape repertoire of the window recogniser, or text was entered with a non-horizontal orientation, a reject was generated. In the set up of the HF experiments, it was not possible for the DAM (and human wizard) to explain why rejects were made. It is expected that such explanations can be provided by subsequent versions of the DAM, leading to a more transparent and understandable system behaviour.

Based on these considerations, the following issues will be examined:

- *In which cases did PII falsely accept user input?* By considering these cases, control parameters such as thresholds can be adjusted, or it may be decided to develop new algorithms that should be able to handle the wrongly recognized inputs, based on the observation that users generate certain shapes that are not considered until now.
- *In which cases did PII wrongly reject user input?* Similar considerations as above hold, possibly leading to conclusions about wrong parameter settings or faulty or incomplete heuristics.
- *What is the recognition performance of PII?* Based on a careful analysis of the loggings, all cases where the user used the pen to enter information can be judged as correct or false. Based on this information, for each user and each session, the recognition performance can be computed. This information is extremely useful to correlate user acceptance rates with recognition performance.

Next to these module-specific research questions, the subjective information provided by the subjects will be analysed to assess the satisfaction rate and usability of “the pen and LCD-tablet” as interaction device.

B.4 The natural language processing module

The task of the natural language processing module is to analyze the output of the speech recognizer (ASR) and the hand writing recognizer. In the case of ASR the output consists not only of the best hypothesis but a scored n-best list is provided. The result of the analysis process is a list of semantic hypotheses describing the user intention in an abstract representation. The abstract representation is based on the system-wide used ontology. Several alternative results are possible since the speech recognizer produces a n-best list and the content of the utterances may be ambiguous leading to different representations.

To support the selection of one result later in FUSION, the different results are scored by the parser using internal semantic knowledge and the expectations provided by DAM. The expectations of DAM describe what the user is expected to do next, e.g., answering a question of the system or selecting an item from a list shown on the display. The expectations also influence the produced output structure ensuring that they are presented in an unique manner independent of the kind of user utterance, e.g., short or more elaborate.

B.4.1 Approach

For the experiments we used an enhanced version of the parser already integrated in the T12 demonstrator. The parser is a template-based semantic parser developed at DFKI for various projects.

As typical for a semantic parser the approach does not include a syntactic analysis, but the high level output structure is build up directly from word level. This is feasible since the input consists of spoken utterances intended to interact with a computer system and therefore, they are syntactically less complicated and limited in length. Furthermore, the lack of a syntactical analysis increases the robustness against speech recognition error.

The used parser differs from other existing approaches by providing a more powerful rule language. Main motivations therefore are to simplify the writing and maintenance of rules and increased flexibility for the output structure. As the amount of rules is getting quite large during the development of a dialog system easy creation and maintenance of the rules is one of the most important issues for parsers in dialog systems.

Several off-line optimizations still provide fast processing despite the increased rule power. The most important optimization is a fixed application order of the rule set with the objective to avoid wasting time by the generation of sub-optimal results.

B.4.2 Research focus

The collected data will be used to enhance the NLP module in two aspects. First, the subjects will utter new speech variants which are not covered by the current knowledge bases. The collected data will be used to extend the knowledge bases accordingly in the future. Second, the principle limits of the scoring function will be examined and also possible enhancements of the scoring function will be tested to provide a better overall performance.

B.5 The Fusion Module

The task of the Fusion module is to combine and integrate the output of the different modalities into a single representation describing the user intention. During processing the module receives different hypotheses containing analyzed output from the gesture recognition module and the natural language processing module. Each input structure has to be interpreted with respect to the context provided by the other modality. Additionally, it is important to take the current dialog state into account. One important task of the FUSION component is related to the synchronization of the output of the different analyzers.

B.5.1 Approach

In contrast to the T12 demonstrator we used for the HF experiments a new approach based on a production rule system. This approach permits a more flexible interpretation and integration of the output received from the different modalities. In contrast to the approach of the NLP module (see section B.4) this approach focuses on a higher level representation to control the application of production rules. This higher level representation consists of two main parts:

(i) a *goal stack* representing the focus of attention in a hierarchical order

(ii) an activation process determining the accessibility of objects in the working memory and of productions in the procedural memory.

Production rules consists of a set of conditions and an action part. The conditions specify the state of the current goal and parts of the working memory, whereas the action part can change the goal stack (either push or pop entries), create new working memory entries, delete no longer required ones, or trigger output of a particular working memory entry. A conflict resolution process selects one production out of the set of applicable productions and executes its action part. The applicability of an production rule is determined by their conditions. When all conditions are fulfilled with respect to a given configuration of the working memory a production rule can possibly be applied. Before the conflict resolution process can select the *firing* production rule, two steps take place to generate the conflict set:

(i) for each rule of the rule set it is checked whether its goal conditions matches the current goal on the goal stack and all rules that pass this check form the *goal-conflict set*

(ii) in a second step the remaining conditions are tested with respect to the current state of the working memory and all rules that pass this test form the actual conflict set

If there is more than one production rule in the final conflict set, a scoring algorithm assess each rule with respect to the current configuration of the working memory and the action part of the highest scored production rule is executed.

Important is also that both the scores of the objects in working memory and of the available productions are dynamically updated during the course of the dialog. Each object in the working memory comprises an *activation* value which decreases during time when the object is not used and increases when, for example, it is referenced by the condition of the firing rule. Analogous, each production has a dynamically updated value which is decreases when it is selected to fire and increase otherwise.

Using such a production rule system for the task of modality fusion has the advantage that once the software is implemented and robust changing or adding new functionality can be done by editing the knowledge bases (the production rules). The production rules we implemented so far are based on the idea that the working memory serves a repository for the current dialog state. Every time FUSION receives a message from one of the analyzers this message is incorporated into the working memory via the goal stack. For each anticipated situation that could happen during the course of dialog there are specialized production rules that trigger the appropriate reaction of FUSION.

B.5.2 Detailed research questions for FUSION

As the human factor experiments were designed not only to evaluate the modules but also to reveal faulty assumptions we focused for FUSION on four research issues:

- *In which cases and why did FUSION produce erroneous output?*
- *Will the subjects accept the strict turn-taking protocol?*
- *Do the subjects really show the envisioned multimodal interaction patterns?*
- *Are there any multimodal interaction patterns that are not covered by the production rules?*