The Potsdam Dialogue Corpora
Experiment Handbook

David Schlangen *and* Raquel Fernández
das|raquel@ling.uni-potsdam.de
Universität Potsdam
Germany

V1.0, September 8, 2008

Contents

1. Preface

This report documents the experiments we conducted in 2006 / 2007 as part of the EU-funded project "DEAWU: Dealing With Uncertainty in Dialogue" (EU Marie Curie Programm 'Transfer of Knowledge') and the DFG-funded project "INPRO: Incrementality and Projection in Dialogue" (Emmy Noether Program). There were four sets of experiments, with the uniting general theme of studying dialogue behaviour under the presence (or absence) or certain restrictions / disruptions. (A more detailed overview can be found in the next chapter.)

The aim of this report is a) to document the annotated corpora that resulted from the experiment, so that it can be used for further analysis or processing, and b) to give enough information to make our results published in (Fernández, Lucht, Rodríguez & Schlangen 2006, Fernández, Schlangen & Lucht 2007, Schlangen & Fernández 2007*b*, Schlangen & Fernández 2007*a*, Fernández & Schlangen 2007) reproducable. Note that the results themselves are not repeated here; for those consult the referenced publications.

Authorship is distributed as follows: This document was written and edited by Raquel Fernández and David Schlangen. Manfred Stede and Andrea Corradini contributed to early discussions on the experimental tasks. The experiments described in Chapters 3, 4 and 5 were initially designed by David Schlangen, with contributions by Raquel Fernández, who supervised the execution of the experiments. The version described in Chapter 7 was developed and run by Alexander Siebert. Transcription was overseen by the authors, with contributions as acknowledged in the chapters below.

Sub-Corpus	Task	Inter-activity	Noise	1stP	Visual	Language
ptt:ptt	pent (s1)	p	×	×	×	German
ptt:ftt	pent (s1)	f	×	×	×	German
NP:no noise	pent (s2)	f	×	√	×	English
NP:noise	pent (s2)	f	√	√	×	English
ND:no noise	dict	f	×	√	n/a	English
ND:noise	dict	f	√	√	n/a	English
visPent	pent (s1)	f	×	×	√	German
RecInstr	pent (var.)	none	×	n/a	n/a	German

Table 1: Overview of available (sub-)corpora

2. Introduction

All experiments had in common that the participants were asked to solve a task together; hence, all recorded dialogues fall in the class of *task-oriented dialogue*. All tasks were asymmetric w.r.t. the participants' knowledge; hence there were recognisable *roles* that were fixed. In the following, we will call these roles *Instruction Giver* (IG) and *Instruction Follower* (IF).

Some parameters varied between the experiments and some were kept constant, as can be seen in Table 1. The codes for the (sub-)corpora are as follows:

- `ptt` denotes an experiment where our main task, the reconstruction by IF of a puzzle to which only IG had a solution, was done via verbal instructions only. (I.e., there was only an audio channel connecting IG and IF.)

  For half of the pairs, the interactivity of the audio channel was restricted by an imposed *push-and-hold-to-talk* policy that allowed only one participant to speak at a time. This is the subcorpus `ptt:ptt`; respectively, `ptt:ftt` (for "free turn taking")) denotes the subcorpus of dialogues where there was no interactivity restriction.

- `NP` denotes an experiment where the same task (albeit with a slightly different solution and start state, where one piece was already placed) was done again via audio channel only, but here not with restrictions on interactivity but with intermittent, randomly placed noise in the channel from IG to IF. This is the subcorpus `NP:noise`; `NP:no noise` is the control condition with no articial noise in the channel.

- The `ND` experiment was run with the same conditions as described for

NP, but a different task. Here IG dictated items read off a computer screens to IF, who typed them out.

- visPent returned to the setting of ptt, but without interactivity restrictions and adding a visual channel from IF to IG. (I.e., IG could see what actions IF performed on the puzzle board.)
- RecInstr finally is a derivative of the puzzle setting in that here all interactivity was removed: a puzzle scene was shown on a computer screen and the human subject was asked to record instructions. These instructions were later played to other subjects, who tried to follow them. I.e., here the connection between IG and IF was severed and interactivity reduced to zero.

We now describe the experiments in detail.

3. Disruption with Noise: Pentomino Task (NP)

This section describes the "Noise Pentomino (NP)" experiment. If you're only interested in the file structure of the corpus distribution, jump ahead to Subsection 3.6.

3.1. Design

In this experiment, two participants solved a puzzle reconstruction task (see Figure 2), with one participant in possession of a solution (instruction giver, IG) and the other participant executing the actions (instruction follow, IF).

The independent variable in this experiment was "presence of noise in channel IG→IF", with two levels "yes, no". Our initial interest was in the number and type of occurences of so-called "clarification requests" as dependent variables; however, we also studied other characteristics of dialogue shape. (For details, see the publications referenced above.)

3.2. Participants

Subjects were recruited through a public call for participation, first in Berlin, Germany and later in London, UK. They were all native English speakers from a variety of native countries. Half of them were college students while the other half had a range of different occupations (including web designers, teachers, musicians and waiters). A full list of participants, with details on age, origin, education, and ocupation is given in Appendix A.1.

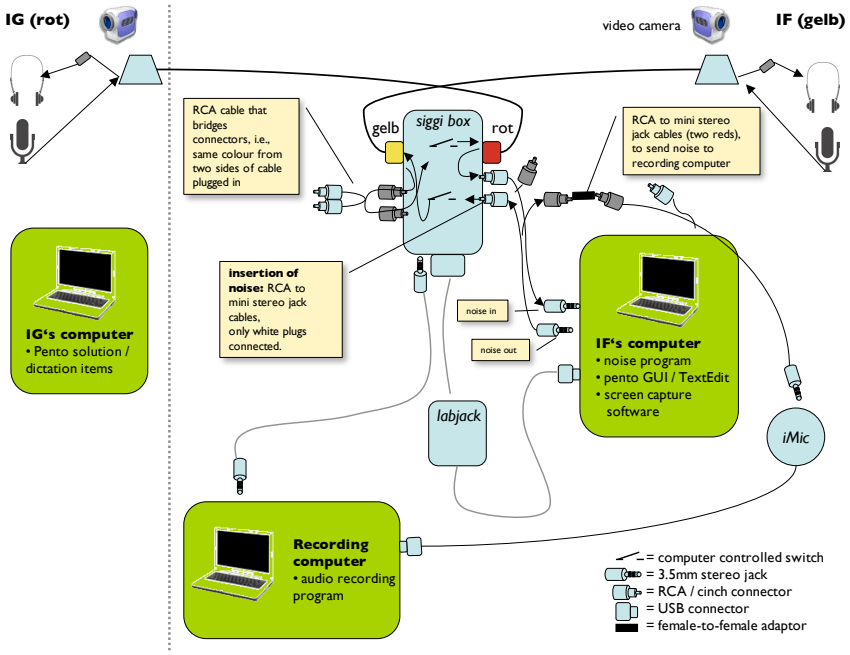


Figure 1: Experimental setup

3.3. Materials / Apparatus

3.3.1. Setup

The diagram in Figure 1 gives an overview of the setup used for the data collection. The setup includes the following material:

- Three computers:
  1. The **IG computer** is used to display the numbered Pentomino solution (see Appendix A.4).
  2. The **IF computer** runs the following software:
     - The Pentomino program (see below). See Figure 2.
     - The screen-capture program *Snapz Pro X (v.2.0.3)*, used to videotape the computer screen.
     - The Noise program (see below)

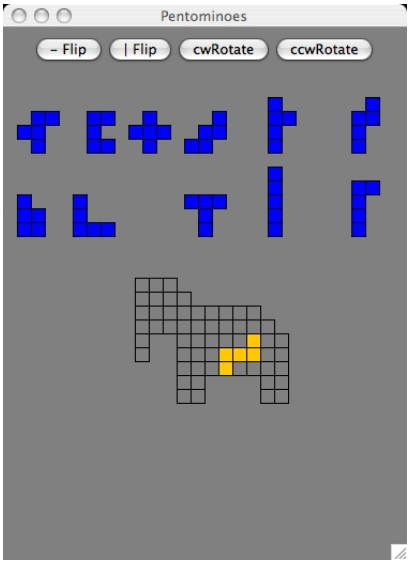


Figure 2: Initial state of Pentomino program

  - TimeStamper (see below)
  - Beeper (see below)

3. The **Recording computer** is used to record the audio. The recording was done with the Apple software *GarageBand (v.2.0.1)*. In control runs, we recorded 2 channels (IG and IF). In the runs with noise, 3 channels were recorded: IG without noise (i.e., IG said), IG with noise (i.e., what IF heard of IG), and IF. Besides the built-in audio inputs of the computer, we used a Griffin iMic USB-audio adapter for the additional third channel.

- The *Siggi Box* – a hardware that allows the connection of two headsets with headphone and microphone (22kHz frequency range) with computer controlled audio-routing. The box is used to route the audio of the IG channel to the Noise program.
- One or two video cameras to videotape the face of the IF and possibly the face of the IG as well.

3.3.2. Software

All software used in the experiment is available in the directory `Software/RunningExp`.
For additional information, see the `README.txt` files for each program.

- *ts_server.perl* is a simple UNIX Socket Server that accepts incoming connections and writes all (newline delimited) input it receives on a socket into a file, together with a time stamp. In the `NP` setup, this script is used to record in a synchronised fashion messages from the programs described below.
- The noise program in the default setting routes audio from the audio input to the audio output. It measures audio activity (i.e., performs simple voice activity detection) and randomly (but constrained by user controllable parameters) replaces signal by noise (of controllable characteristic).

  We used the following setting for our experiments:

  ```
  noisychannel -mode 2 -noise b -nl 0.3 -t -100 -f 100 -np 0.01 -sp 0.2 -nd 100
  ```

  The program logs each event (start/end of signal replacement) to a file; in our experiment, this log was piped to the timestamping program.
- The puzzle itself—a well-known game called *Pentomino*—was realised by a small Java program (see Figure 2), which was originally written Robert Simms and modified for the current purposes by David Schlangen. The program writes out information about the performed actions (pieces selected & manipulated) to STDOUT, which we piped to the timestamper to be integrated into the main logfile.

  This information can be used to "replay" a run (albeit without mouse movements); the necessary software to do this is included in the directory.
- The *BeeperScript* finally simply displays a message on the screen (which is relayed to the timestamper) and plays a synchronised audio signal; it is used to be able to align the screen video (where the message displayed on the screen is shown) and the audio recording (where the beep is heard).

3.4. Procedure

Each experimental run involved two subjects, who first tackled the Pentomino task and then the Dictation task (see Section 4).

Subjects were jointly greeted by the experimenter, who briefly explained the tasks to be carried out and allowed them to choose their roles as either *instruction giver* (IG) or *instruction follower* (IF). Subjects were given a consent form, which they were asked to sign if they agreed.

IF and IG were then placed in different sound-proof rooms and were given written instructions for the Pentomino task. The instructions can be found in Appendix A.2. The IF was allowed a few minutes to get used to the Pentomino program (see Figure 2).

After subjects had read the instructions, the experimenter asked to each of them whether they had any questions. Before leaving the IF room, the experimenter said to the IF something to the effect of: "There might be some problems with the audio, which we can't fix at the moment, so please just go ahead". This was done in order to prevent the subject from coming out of the room to complain about the quality of the audio. Finally the experimenter left the two rooms and the first phase of the run began.

Once the Pentomino task was finished, the experimenter asked the subjects to fill in a questionnaire (see Appendix A.3).

3.5. Analysis

Transcription and Annotation are described in the separate "Transcription and Annotation Handbook", which should be in the same directory as this document.

3.6. Available Data

3.6.1. File Naming Conventions

The audio / video data is sorted by dialogues, and stored in directories named in the format `YYYY-MM-DD_runX` (e.g. `2007-01-24_run1`, where `2007-01-24` is the date of the data collection and `run1` the dialogue identifier within that experimental session). See next section for the path within the main corpus directory. Each dialogue directory contains the following files, where `condition` can be either `noise` or `nonoise`:

- `YYYYMMDD_runXpento_condition_split_IG.aif`
  A one-channel (mono) audio file with the IG only (before noise, if belonging to `noise` condition; i.e., what really was said).
- `YYYYMMDD_runXpento_condition_split_IGnoise.aif`
  A one-channel (mono) audio file with the IG after noise (i.e., what IF heard); only present in the `noise` condition.

- `YYYYMMDD_runXpento_condition_split_IF.aif`
  A one-channel (mono) audio file with the IF only.
- `YYYYMMDD_runXpento_condition.aif`
  A two-channel (stereo) audio file where one channel corresponds to the IF and the other one to the IG. In the `noise` condition, the IG channel is a mixdown between the audio before noise and after noise.
- `YYYYMMDD_runXpento_condition_face_IG.mov`
  A one-channel (mono) QuickTime movie of the face of the IG (and audio captured by the camera).
- `YYYYMMDD_runXpento_condition_face_IF.mov`
  A one-channel (mono) QuickTime movie of the face of the IF (and audio captured by the camera).
- `YYYYMMDD_runXpento_condition_screen.mov`
  A one-channel (mono) QuickTime movie of the Pentomino program on the IF's screen synchronised with audio from `YYYYMMDD_runXpento_condition.aif`
- `YYYYMMDD_runXpento_condition_events.log`
  A log of time-stamped noise events and total noise and signal probability; only present in the `noise` condition.
- `YYYYMMDD_runXpento_condition_mlog.txt`
  A log of time-stamped noise events (if in noise condition), Pentomino events, and 'beeps'.

3.6.2. Corpus Contents

The corpus contains the following dialogues.

Noise condition. Directory path:
`Disruption_with_Noise_Corpus/Data/Pentomino_Task/Noise`[1]

Dialogue	Cond.	Comments
20061117_run1	noise	all data available; reconstructed `screen.mov` from Pentomino events in `mlog.txt`
20061117_run2	noise	all data available
20061117_run3	noise	problematic run
20070117_run1	noise	all data available except `face_IG.mov`
20070131_run1	noise	all data available except `face_IG.mov`
20070131_run2	noise	all data available except `face_IG.mov`
20070131_run3	noise	all data available except `face_IG.mov`

No_Noise condition. Directory path:
`Disruption_with_Noise_Corpus/Data/Pentomino_Task/NoNoise`

Dialogue	Cond.	Comments
20061123_run1	nonoise	all data available except `face_IG.mov`
20070118_run1	nonoise	problematic run
20070124_run1	nonoise	all data available except `face_IG.mov` and `screen.mov`
20070124_run2	nonoise	problematic run
20070201_run1	nonoise	all data available except `face_IG.mov`
20070201_run2	nonoise	problematic run
20070201_run3	nonoise	all data available except `face_IG.mov`
20070201_run4	nonoise	all data available except `face_IG.mov`

3.7. References and Acknowledgements

We are grateful to our students Sebastian Bachman, Amaya Steinhilbert, Alexander Siebert, and Hannah Bohle, who helped during some data collection sessions and contributed to the transcription of the dialogues.

Some analyses of the described corpus are presented in (Schlangen & Fernández 2007*b*).

[1] Problematic runs are stored in a sub-directory `ProblematicRuns`. See README file in that directory for details.

4. Disruption with Noise: Dictation Task

4.1. Design

4.2. Participants

The participants who took part in the Dictation task were the same that carried out the Pentomino task. They were all native English speakers. A full list of participants, with details on age, origin, education, and ocupation is given in Appendix A.1.

4.3. Materials / Apparatus

The setup used for the data collection of the Dictation task is equivalent to that shown in Figure 1 and described in Section 3.3.

The materials used only differ in the following aspects:

- The **IG computer** is used in this case to display HTML pages on a web browser that show the items to be dictated one at a time. See Appendix A.5 for a list of all items.
- The **IF computer** is used in this task to run the Apple program *TextEdit (v.1.4)* (instead of the Pentomino program), where the IF types in the dictated items. All other programs run by this computer are as listed in Section 3.3.

4.4. Procedure

The Dictation task was carried out after the Pentomino task in the second half of each experimental run (see Section 3.4). Subjects kept the roles (IG/IF) taken in the Pentomino task.

After the subjects had filled in the Pentomino questionnaire, the experimenter gave them written instructions about the Dictation task (see Appendix A.2). The IG was shown an HTML page on a web browser that displayed the items to be dictated one at a time. The IF was shown a simple text editor and was told to type in the items being dictated. After these instructions, the experimenter left the rooms and the dictation phase of the experimental run began.

Once the Dictation task was finished, subjects were asked to fill in a second instance of the questionnaire (see Appendix A.3).

4.5. Available Data

The data available for each dialogue is stored in a directory named in the format `YYYY-MM-DD_runX` (e.g. `2007-01-24_run1`, where `2007-01-24` is the date of the data collection and `run1` the dialogue identifier within that experimental session). Each dialogue directory contains the following files, where `condition` can be either `noise` or `nonoise`:

- `YYYYMMDD_runXdict_condition_split_IG.aif`
  A one-channel (mono) audio file with the IG only (before noise, if belonging to `noise` condition).
- `YYYYMMDD_runXdict_condition_split_IGnoise.aif`
  A one-channel (mono) audio file with the IG after noise; only present in the `noise` condition.
- `YYYYMMDD_runXdict_condition_split_IF.aif`
  A one-channel (mono) audio file with the IF only.
- `YYYYMMDD_runXdict_condition.aif`
  A two-channel (stereo) audio file where one channel corresponds to the IF and the other one to the IG. In the `noise` condition, the IG channel is a mixdown between the audio before noise and after noise.
- `YYYYMMDD_runXdict_condition_face_IG.mov`
  A one-channel (mono) QuickTime movie of the face of the IG (and audio captured by the camera).
- `YYYYMMDD_runXdict_condition_face_IF.mov`
  A one-channel (mono) QuickTime movie of the face of the IF (and audio captured by the camera).
- `YYYYMMDD_runXdict_condition_screen.mov`
  A one-channel (mono) QuickTime movie of the text editor on the IF's screen synchronised with audio from `YYYYMMDD_runXdict_condition.aif`
- `YYYYMMDD_runXdict_condition.rtf`
  A copy of the RTF file at the end of the task created by the text editor
- `YYYYMMDD_runXdict_condition_events.log`
  A log of time-stamped noise events and total noise and signal probability; only present in the `noise` condition.
- `YYYYMMDD_runXdict_condition_mlog.txt`
  A log of time-stamped noise events (if in noise condition), and 'beeps'.

4.5.1. Corpus Contents

The corpus contains the following dialogues.

Noise condition. Directory path:
`Disruption_with_Noise_Corpus/Data/Dictation_Task/Noise`[2]

Dialogue	Cond.	Comments
20061117_run1	noise	all data available
20061117_run2	noise	all data available
20061117_run3	noise	all data available
20070117_run1	noise	all data available except `face_IG.mov`
20070131_run1	noise	all data available except `face_IG.mov`
20070131_run2	noise	all data available except `face_IG.mov`
20070131_run3	noise	all data available except `face_IG.mov`

No_Noise condition. Directory path:
`Disruption_with_Noise_Corpus/Data/Dictation_Task/NoNoise`

Dialogue	Cond.	Comments
20061123_run1	nonoise	all data available except `face_IG.mov`
20070118_run1	nonoise	problematic run
20070124_run1	nonoise	all data available except `face_IG.mov` and `screen.mov`
20070124_run2	nonoise	problematic run
20070201_run1	nonoise	all data available except `face_IG.mov`
20070201_run2	nonoise	problematic run
20070201_run3	nonoise	all data available except `face_IG.mov`
20070201_run4	nonoise	all data available except `face_IG.mov`

4.6. Analyses

Described in the "Transcription and Annotation Manual".

4.7. References and Acknowledgements

The students Sebastian Bachman, Amaya Steinhilbert, Alexander Siebert, and Hannah Bohle helped during some data collection sessions and con-

[2] Problematic runs are stored in a sub-directory `ProblematicRuns`. See README file in that directory for details.

tributed to the transcription of the dialogues.

Some results of analyses of this corpus are presented in (Schlangen & Fernández 2007*b*).

5. Disruption of Interactivity: Push-to-Talk

5.1. Design

The aim of this experiment was to investigate in isolation the effects of restricting interactivity by using a half-duplex channel managed by *push-to-talk*. The experiment involved two conditions: a fully interactive *free turn-taking* (FTT) condition (control group) and a restricted *push-to-talk* (PTT) condition (experimental group).

The task we asked our subjects to do was the Pentomino task described in 3.1. Note that in the materials and publications related to this experiment the roles of Instruction Giver and Instruction Follower are often referred to as Player and Executor, respectively.

5.2. Participants

A total of 20 subjects participated in the experiment, 11 females and 9 males. They were all German native speakers between 20 and 45 years old. Subjects were grouped in 10 pairs and 5 pairs were assigned to each of the two conditions: two female-female pairs, one male-male pair, and two female-male pairs used FTT, while two female-female pairs, two male-male pairs, and one female-male pair used PTT.

Detailed information on age and gender of participants for each dialogue is given in Appendix B.1.

5.3. Materials / Apparatus

The following equipment was used to carry out the experiment: a purpose-built, cardboard pentomino game consisting of a puzzle board and a set of 12 loose pieces, a digital camera (used to videotape the board during task execution), a mixing desk, and a computer for recording (we used the free software *Audacity*[3] v. 1.2.6 and the Apple program *GarageBand* v. 2.0.1.).

In the FTT condition, 2 microphones and headsets were used, with the audio channel being continuously open in both directions. In the PTT condition, the mics were used to capture the audio, but the headsets were substituted by a set of two AUDIOLINE PMR 012 walkie-talkies. Here speakers have to press a button in order to take the turn, hold it to keep it, and release it again to yield it; a 'beep' is heard by both parties when the turn is yielded.

[3] http://audacity.sourceforge.net/

5.4. Procedure

Participants were placed in two different sound-proof rooms connected by an audio line. They were separately briefed on the task. IGs were given short written instructions. In the PTT condition, subjects were allowed a few minutes to familiarise themselves with the workings of the walkie-talkies.

IGs had in front of them a numbered solution of the puzzle (see Appendix B.2), while IFs had the empty board and the set of loose pieces. Note that in this experiment no initial pentomino piece was placed on the board.

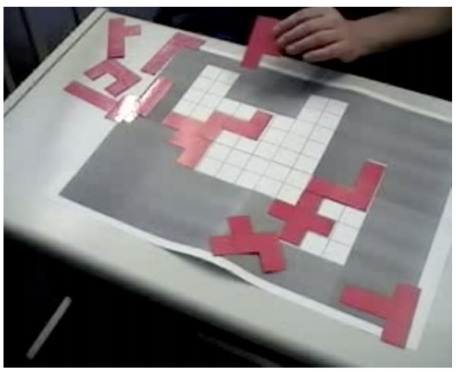
Once the task was finished, participants completed a user questionnaire. The questionnaires (in German) are available in Appendix B.3.

5.5. Available Data

The data available for each dialogue is stored in a directory named in the format `condition_YYYY-MM-DD_pairX` (e.g. `PTT_2006-03-27_pair1`, where `FTT` is the turn-taking condition, `2006-03-27` is the data collection date, and `pair1` is the identifier of the pair of participants in the dialogue.

Each dialogue directory contains the following files:

- `YYYY-MM-DD_pairX.wav` or `YYYY-MM-DD_pairX.aif` A two-channel (stereo) audio file with both sides of the conversation
- `YYYY-MM-DD_pairX.mov` A one-channel (mono) QuickTime movie of the board manipulated by the IF (with the audio captured by the camera).



The corpus contains the following dialogues:

FTT condition. Directory path: `Push_to_Talk_Corpus/Data/FTT`

Dialogue	Comments
FTT_2006-02-13_pair1	only audio; no movie available
FTT_2006-03-27_pair1	all data available
FTT_2006-03-28_pair2	all data available
FTT_2006-06-22_pair1	all data available
FTT_2006-06-22_pair2	all data available

PTT condition. Directory path: `Push_to_Talk_Corpus/Data/PTT`

Dialogue	Comments
PTT_2006-02-13_pair2	all data available
PTT_2006-03-27_pair2	all data available
PTT_2006-03-27_pair3	all data available
PTT_2006-03-28_pair1	all data available
PTT_2006-06-22_pair3	all data available

5.6. Analyses

5.6.1. Transcription

The transcription was done in Praat. Initially (April-June 2006) dialogues were only segmented into turns. At a later stage (April 2007) dysfluencies were marked in the transcripts and turns were segmented into utterances following the guidelines given in the "Transcription and Annotation Manual".

5.6.2. Annotation

Annotations were done in MMAX2. The 2006 transcription was annotated at the level of *moves* and at the level of *dialogue acts* (DAs). For the annotation of the 2007 transcription, markables for turns and utterances were automatically imported from Praat. These transcripts are being annotated at the level of *moves*, *dialogue acts*, and *referential expressions*. DA markables correspond to the automatically imported markables at the utterance level. Markables for the other two levels are created during annotation.

The schemas are described in the "Transcription and Annotation Manual".

5.7. References and Acknowledgements

The experiment was conducted in the Phonetics Lab at the Zentrum für Allgemeine Sprachwissenschaft (ZAS) in Berlin, for which we are grateful to Jörg Dreyer. Tatjana Lucht was the transcriber of the dialogues and the annotator of the 2006 transcripts. The annotation of the 2007 version of the transcription was done by Janine Wolf.

Results of this experiment have been published in (Fernández et al. 2006, Fernández, Corradini, Schlangen & Stede 2007, Fernández & Schlangen 2007)

6. The VisPento Corpus

The setting for the visPento experiments was much like that for noise pento, except that there was a visual channel between IG and IF in that IG could see the actions on the board performed by IF. This was realised technically through a VNC connection between the IF computer and a computer in IG's room. Recording was done as described for noise pento, no-noise. The game board used was the same as in the PTT experiments, i.e., with no pre-placed piece.

7. The Pento Naming Corpus

The `PentoNamingCorpus` is a variant of the pentomino setting, but with only one participant. The participant was presented with half-completed outlines and unplaced pieces. One of the unplaced pieces was highlighted, and a corresponding location on the outline, and the participant was asked to record instructions for performing one move.

The recordings were performed in a distributed fashion as described in (Siebert, Schlangen & Fernández 2007). In that way, for each recorded instruction there is also an evaluation in terms of how difficult it was to follow it. 300 scene descriptions were recorded in this way, by 10 speakers. An additional 300 scene descriptions were later recorded with one speaker.

All material from this data collection can be found in the directory *PentoNamingCorpus*. (Siebert & Schlangen 2008) builds on this data.

Appendices

A. Disruption with Noise

A.1. Participants

Details for each participant include: gender, age range, origin, highest education, and current ocupation.

Participants in the NOISE condition

Run	Role	Details
20061117_run1	IG	male, 30/40, England, BA in Behavioural Science, teacher
	IF	male, 20/30, Canada, Bachelor of Architecture, filmmaker
20061117_run2	IG	female, 40/50, USA, 1 year college, performer/designer
	IF	male, 30/40, New Zealand, Masters, web designer
20061117_run3	IG	female, 30/40, Romania, Diploma, journalist/English teacher
	IF	male, 20/30, Scotland, BA, weiter
20070117_run1	IG	female, 20/30, Ireland, Postgraduate Diploma, writer/bartender
	IF	female, 20/30, England, Masters, unemployed
20070131_run1	IG	male, 20/30, England, A levels, college student
	IF	male, >20, England, A levels, Law student
20070131_run2	IG	male, 20/30, United Kingdom, A levels, college student
	IF	male, 20/30, England, GCSE, student/kitchen assistant
20070131_run3	IG	female, 20/30, England, Master's, postgraduate student
	IF	female, 30/40, USA, Masters, PhD student

Participants in the NO_NOISE condition

Run	Role	Details
20061123_run1	IG	female, 30/40, USA, BFA, photographer/artist
	IF	female, 40/50, USA, 2 years college, musician/baker
20070118_run1	IG	male, >20, England, A levels, student
	IF	male, >20, England, A levels, student
20070124_run1	IG	male, 30/40, USA, Bachelor of Science, English teacher
	IF	female, 30/40, USA, Bachelor of Arts, furniture designer
20070124_run2	IG	male, 20/30, Sierra Leone, Master's, student
	IF	male, 20/30, England, BA, student
20070201_run1	IG	male, 20/30, England, A levels, college student
	IF	female, 20/30, United Kingdom, 1 year Msc, student
20070201_run2	IG	male, 20/30, Iran, MSc, PhD student
	IF	female 30/40, United Kingdom, GCSE, student
20070201_run3	IG	male, 20/30, England, A levels, college student
	IF	male, 20/30, England, A levels, college student
20070201_run4	IG	male, 20/30, Britain, BA, Master's student
	IF	male, 20/30, India, Bsc, college student

A.2. Instructions Given to Participants

Task: Pentomino, Role: IG

```
You are given the solution of a puzzle made up of a set of pieces
put together to form a figure. The task to be done consists in telling
the other participant how to build the puzzle.

The setting available to the other participant is the following.

A screen shows:

     - a gridded outline of the figure with one piece already on it
     - the other pieces in random positions outside the outline. The
       pieces on the side of the other participant are *not* numbered
       and they all have the same colour.

The other participant can rotate and flip around their pieces.

The pieces in the solution of the puzzle you are given are numbered.
These numbers indicate the order you must follow in your instructions
to the other participant on how to build the figure. Start with
piece 1, then move on to piece 2, and so on until you reach piece 11
and the other participant has solved the puzzle.

One of the pieces in the solution is not numbered. This piece is
already placed on the right position on the gridded outline available
to the other participant. Again, keep in mind that the pieces on their
side are *not* numbered and all have the same colour.

If you have any questions, please ask us before the experiment starts.

Thanks again for participating in the experiment.
```

Task: Pentomino, Role: IF

```
On a screen, you are given the outline of a puzzle and a set of pieces.
The other participant has the solution of this puzzle. The task to be
done consists in the other subject telling you how to fill in the
outline with the pieces, following a particular order of the pieces.

The pieces on the screen can be selected by clicking on them. Once
they are selected, they can be rotated and flipped by clicking on the
corresponding buttons:

- Flip    => flipping vertically, on the vertical axis (left <-> right)
| Flip    => flipping horizontally, on the horizontal axis (up <-> down)
cwRotate  => clockwise rotation, 90 degrees
ccwRotate => counter clockwise rotation, 90 degrees

To place a piece on the grid, select it and drag it to the desired
position. When a piece matches a grid position its colour changes.

Pieces can be rotated/flipped only when they are not on the grid.
This means that if you have placed a piece on the grid and then want
to rotate/flip it, you will need to take it out of the grid first.

Please take a moment to familiarise yourself with the program.

One of the pieces will already be placed on the right position in
the gridded outline.

During the experiment, all you need to do is follow the instructions
given by the other subject; you can talk freely during the process.

If you have any questions, please ask us before the experiment starts.

Thanks again for participating in the experiment.
```

Task: Dictation, Role: IG

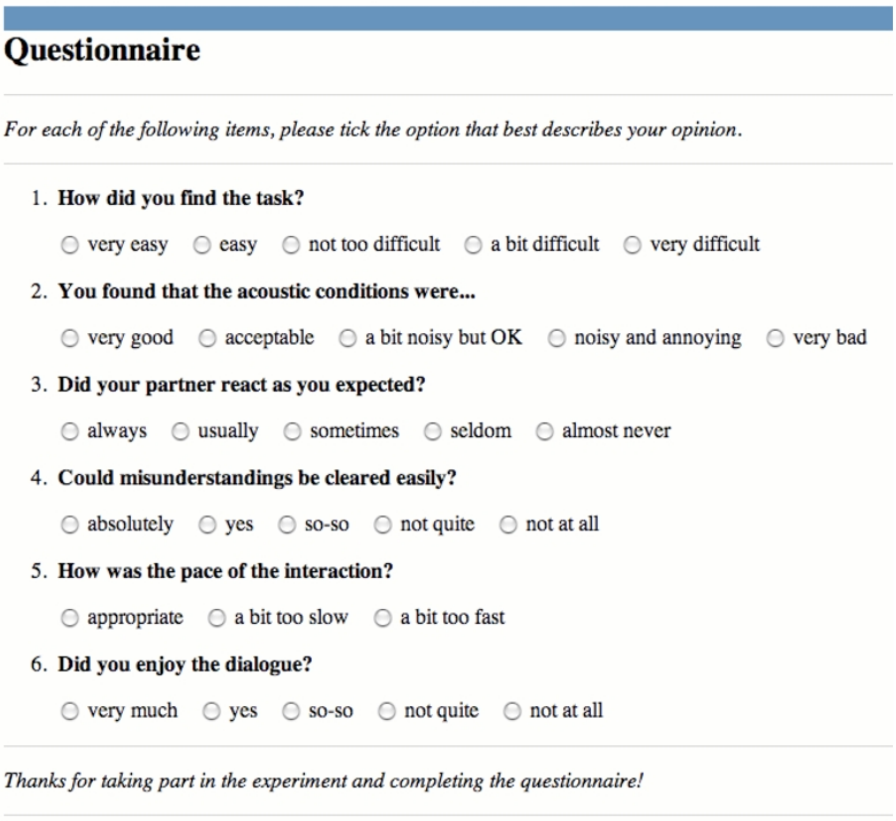
```
The task to be carried out consists in dictating several sentences
and strings of numbers to the other participant, who will type them
in on a computer.

On your computer screen, you will see a window with one of the items
to be dictated. For instance:

     129 485 246 293 832
or
     John loves Mary

Once the other participant has typed in the item on their side,
clicking on the 'Next' button at the bottom of the window will take
you to the next item.

Please do not go back once you have pressed the 'Next' button.
Just go on at your own pace until no further items appear.

If you have any questions, please ask us before the experiment starts.

Thanks again for participating in the experiment.
```

Task: Dictation, Role: IF

```
On the screen you will see a text editor. The task to be carried out
consists in typing in the sentences and strings of numbers that the
other participant will dictate to you. For instance:

     129 485 246 293 836
or
     John loves Mary

Please make an effort to type in *exactly* what the other participant
dictates to you.

Press 'return' after each item (sentence or string of numbers),
and please don't move the text editor around!

Note that the other participant sees one of the items to be dictated
at a time. Once you have decided to move on to the next sentence or
string of numbers there is no way to go back.

If you have any questions, please ask us before the experiment starts.

Thanks again for participating in the experiment.
```

A.3. Questionnaire

The following questionnaire was filled in by participants after finishing the Pentomino and the Dictation tasks, respectively. Answers to the questionnaires are stored in the directory `Disruption_with_Noise/QuestionnaireResults`.



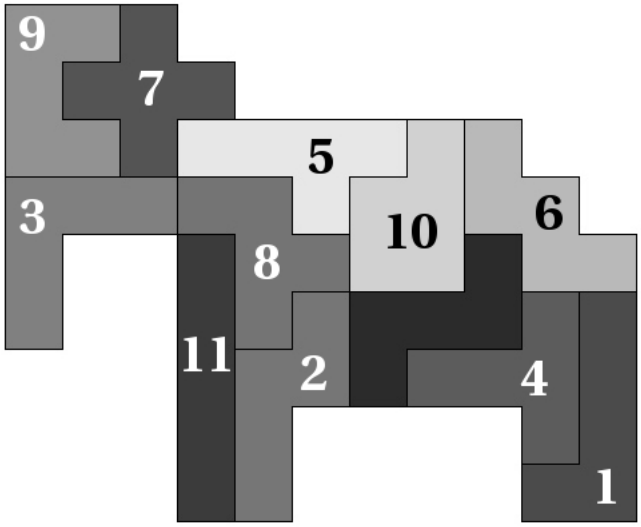
Questionnaire

For each of the following items, please tick the option that best describes your opinion.

1. **How did you find the task?**
   ○ very easy ○ easy ○ not too difficult ○ a bit difficult ○ very difficult
2. **You found that the acoustic conditions were...**
   ○ very good ○ acceptable ○ a bit noisy but OK ○ noisy and annoying ○ very bad
3. **Did your partner react as you expected?**
   ○ always ○ usually ○ sometimes ○ seldom ○ almost never
4. **Could misunderstandings be cleared easily?**
   ○ absolutely ○ yes ○ so-so ○ not quite ○ not at all
5. **How was the pace of the interaction?**
   ○ appropriate ○ a bit too slow ○ a bit too fast
6. **Did you enjoy the dialogue?**
   ○ very much ○ yes ○ so-so ○ not quite ○ not at all

Thanks for taking part in the experiment and completing the questionnaire!

A.4. Numbered Pentomino Solution



A.5. Dictation Items

- **Strings of numbers**

```
875 820 937 676 493
967 298 455 629 184
572 467 261 933 931
109 576 276 349 857
644 189 265 708 312
917 845 418 991 296
675 512 802 347 534
409 573 245 371 245
822 305 726 425 179
456 829 856 102 984
```

- **Sentences**

```
John came a long way from Manchester to deal with that.
Rather than directing, be directive.
There are some good reasons why we need this skill.
The importance of the game has faded to nothing
The blinds on the third floor are drawn
They frequently lean on each other's shoulders
Guess where Tom went on his way from the station
There is no reason not to carry this through
Currently only males can ascend the Japanese throne
A town in northern Uganda is benefiting from peace efforts across the border.
Improvements in security have helped increase cross-border commerce
There are subsistence farmers that have no strategic plan for farming.
Traders and transporters alike complain of the state of the roads
This organization also hosts a range of other projects
The future may be a little brighter for forests
Making promises and setting goals that are unrealistic is bad
The diagram shows changes in wood volume
```

- **Idioms**

```
A stitch in time saves nine
This rule applies across the board
Scientists have been barking up the wrong tree
It was easy to read between the lines
That's easier said than done
A matter of life and death needs careful thought
John's phone call killed two birds with one stone
Don't judge a book by the cover
Everybody gets out of bed on the wrong side sometimes
This computer costs an arm and a leg
```

- **Modified idioms**

```
Jim was a big fish in a deep pond
All doors lead to Rome
The president was caught soft-handed
In this department the left hand doesn't know what the small hand is doing
Cris had the right word on the base of his tongue
Children always know which side their head is buttered on
You can't get plums out of a stone
```

A.6. Technical Setup of Experiments

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Internal Instructions / Checklists for setup:


Noise, Checklist:
>>>>>>>
NOISE CONDITION
=====================================================

PREPARATION

 - Set up hardware for noise condition:
        - like in the diagram with labjack unplugged
        - select LP mode in the camera


- In Executor's room:

  - Recording Computer.
       - Applications->Utilities->Audio MIDI Setup:
       'Default Input' and 'Properties for' = 'Aggregate Device'
        'Default Output' and 'System Output' = 'Built-in Audio'

       - Garage Band:
      - Preferences->Audio/MIDI,
       Audio Output = 'Built-in Audio' and Audio Input = 'Aggregate Device'

      - create 3 tracks: channel 1, channel 2, and channel 4
          - lower the bpm setting to minimum

  - Noise Computer

- system preferences->audio, 'audio in'='line in', 'audio out'='line out',

- volume of computer turned to max

- open 5 terminal windows and go to NoiseExpSoft directory
    t1 -- master logger
    t2 -- noise program
    t3 -- tail noise events to master log
    t4 -- beeper     cd Beeper
        t5 -- GUI
- open CopyPaste.txt

  - Test recording (following steps below to start a run)

  - kill processes in Noise Computer
```

```
  - remove events.log
  - remove RUN_mlog.txt


- In Player's room, open solution in Player's computer

-----------------------------------------------------------------------

INSTRUCT SUBJECTS

- Give instructions to Executor
      - start GUI in Noise Computer
      - let the Executor familiarise with the GUI
      - kill GUI

-Give instructions to Player, show solution on screen and example of GUI

-----------------------------------------------------------------------

TO START A RUN

using CopyPaste.txt:

1 - start master logger in t1 Noise Computer
2 - start GUI and tail to master logger in t5 Noise Computer
3 - start noise program in t2 Noise Computer
4 - tail noise events to master logger in t3
5 - start recording with Garage Band in Recording Computer
6 - place beeper terminal t4 on top of GUI and start screen capture
7 - run the beeper in t4 Noise Computer twice

7 - hit record in camera

-----------------------------------------------------------------------

TO FINISH A RUN

1 - stop recording in Recording Computer and save audio in GB format
2 - stop screen capture
3 - stop cameras

4 - stop noise program in t2 Noise Computer
5 - kill processes on terminal windows and kill terminal in Noise Computer
6 - kill GUI in Noise Computer

7 - rename RUN_mlog.txt and events.log created in Noise Computer as
       year-month-day_runX_noise_mlog.txt
   year-month-day-runX_noise_events.log
       (where X is the run number) and move to asafe location
```

```
-----------------------------------------------------------------

- User Questionnaires (fill in run number, noise condition and subject role)

[don't do the following if wizard data collection comes next]

- Consent forms
- Payment and signature
- Debrief subjects: tell them what the experiment was about and get informal feedback

-----------------------------------------------------------------

IMMEDIATE POST-PROCESSING

- Save GUI movie as year-month-day_runX_noise.mov

- Export Audio
   - create one stereo file from the three tracks recorded
       - the right channel (R) for the executor
       - the left channel (L) for the player-with-noise AND the player-no-noise
       - regulate the volume so that the player-without-noise is audible
  - export the file: File->Export to iTunes
  - recover the file from iTunes, rename as year-month-day_runX_noise.aif
     (where X is the run number) and move to safe location

-----------------------------------------------------------------

DATA OBTAINED WITH EACH RUN:

1 audio file: year-month-day_runX_noise.aif
1 master log file:  year-month-day_runX_noise.txt
1 events log file: year-month-day-runX_noise_events.log
1 GUI movie:        year-month-day_runX_noise.mov
1 digital video


----

Checklist NoNoise:

NO-NOISE (TELEPHONE) CONDITION
==================================================

PREPARATION

 - Set up hardware for no-noise (telephone) condition
        - bridge both sides of siggi box with labjack unplugged
```

```
- In Executor's room:

  - Noise Computer only runs the GUI, master logger and SnapzPro

  - Recording Computer.
       - Applications->Utilities->Audio MIDI Setup:
       'Default Input' and 'Properties for' = 'Aggregate Device'
            'Default Output' and 'System Output' = 'Built-in Audio'

       - Garage Band:
      - Preferences->Audio/MIDI,
       Audio Output = 'Built-in Audio' and Audio Input = 'Aggregate Device'

      - create 2 tracks: channel 1 and channel 2

    - open 4 terminal windows and go to NoiseExpSoft directory
      t1 -- master logger
      t2 -- GUI
      t3 -- tail GUI events to master log
      t4 -- beeper     cd Beeper

- open CopyPaste.txt

  - Test recording (following steps below to start a run)

  - kill processes in Noise Computer
  - remove events.log
  - remove RUN_mlog.txt

- In Player's room, open solution in Player's computer

----------------------------------------------------------------------

INSTRUCT SUBJECTS

----------------------------------------------------------------------

TO START A RUN

using CopyPaste.txt:

1 - start master logger in t1 Noise Computer
2 - start GUI in t2 and tail to master logger in t3 Noise Computer
3 - start recording with Garage Band in Recording Computer
4 - place beeper terminal t4 on top of GUI and start screen capture
5 - run the beeper in t4 Noise Computer twice

6 - hit record in camera
```

```
---------------------------------------------------------------------

TO FINISH A RUN

1 - stop recording in Recording Computer and save audio in GB format
2 - stop screen capture
3 - stop cameras

---------------------------------------------------------------------

- User Questionnaires (fill in run number, noise condition and subject role)
- Consent forms, Payment and signature
---------------------------------------------------------------------

IMMEDIATE POST-PROCESSING

- Export Audio
   - create one stereo file from the two mono tracks recorded
       - the right channel (R) for the executor
       - the left channel (L) for the playe
  - export the file: File->Export to iTunes
  - recover the file from iTunes, rename as year-month-day_runX_no-noise.aif
     (where X is the run number) and move to a safe location

---------------------------------------------------------------------

DATA OBTAINED WITH EACH RUN:

1 audio file: year-month-day_runX_no-noise.aif
1 master log file:  year-month-day_runX_noise.txt
1 GUI movie:        year-month-day_runX_noise.mov
1 digital video
-----------------



CopyAndPaste.txt

==================================
NOISE CONDITION
==================================

COMMANDS TO BE COPIED INTO 4 TERMINAL WINDOWS IN NOISE COMPUTER

1 - Master Log
--------------

tcsh
```

```
TimeStamper/ts_server.perl RUN_mlog.txt &
tail -f RUN_mlog.txt


2 - GUI tailed to master log
----------------------------

java -jar das_pentomino.jar | nc localhost 9000


3 - Noise Program
-----------------

tcsh
noisychannel_v1.0bRC2/mac_osx/noisychannel -mode 2 -noise b -nl 0.3 -t -100 -f 100 -np 0.01 -sp 0.2 -nd 100


4 - Noise events to Master Log
------------------------

tail -f events.log | nc localhost 9000


**** START RECORDING AUDIO WITH GARAGE BAND****

4 - Beep
---------

./beep.perl



====================================
WIZARD EXPERIMENT
====================================

COMMANDS TO BE COPIED INTO 5 TERMINAL WINDOWS IN WIZARD COMPUTER

1- Master Log
------------

TimeStamper/ts_server.perl RUN_mlog.txt &
tail -f RUN_mlog.txt


2- Wizard UI
-----------

wish wizardUI.tcl > RUN_wiz.log &
```

```
tail -f RUN_wiz.log | nc localhost 9000


3- PTT controller
--------------

./labjack > RUN_sbox.log &
tail -f RUN_sbox.log | nc localhost 2455


4- Wizard GUI
------------

java -jar das_pentomino.jar | tee RUN_pent.log | nc <name_of_user_machine> 9123


5- GUI log to Master Log
--------------------

tail -f RUN_pent.log | nc localhost 9000
```

B. Push-to-talk

B.1. Participants

The follwoing tables list the participants who took part in the data collection.
They were all German native speakers. Details include gender and age.

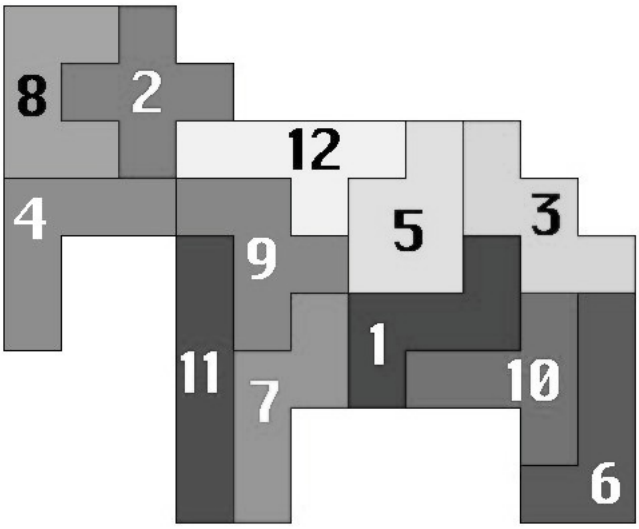
Participants in the FTT condition

Run	**Role**	**Details**
2006-02-13_pair1	Player	female, ∼ 35
	Executor	female, 32
2006-03-27_pair1	Player	male, 40
	Executor	male, 40
2006-03-28_pair2	Player	male, ∼ 28
	Executor	female, ∼ 25
2006-06-22_pair1	Player	male, ∼ 25
	Executor	female, ∼ 25
2006-06-22_pair2	Player	female, ∼ 23
	Executor	female, ∼ 23

Participants in the PTT condition

Run	**Role**	**Details**
2006-02-13_pair2	Player	male, ∼ 40
	Executor	male, ∼ 45
2006-03-27_pair2	Player	female, 28
	Executor	female, 28
2006-03-27_pair3	Player	male, 47
	Executor	female, 33
2006-03-28_pair1	Player	female, 35
	Executor	female, 25
2006-06-22_pair3	Player	male, ∼ 30
	Executor	male, ∼ 28

B.2. Numbered Solution



B.3. User Questionnaires

B.3.1. Questionnaire for Instruction Givers (Players)

```
1) Wie bewerten Sie den Schwierigkeitsgrad der Aufgabe?
[ ] einfach [ ] schwerer als erwartet [ ] schwer [ ] nicht machbar

2) Wie viel Zeit haben Sie f\"ur die L\"osung der Aufgabe beansprucht?
[ ] sehr wenig [ ] weniger als erwartet [ ] mehr als erwartet
[ ] sehr viel Aufgabe war nicht l\"osbar

3) Wo lag f\"ur Sie die gr\"osste Schwierigkeit beim L\"osen der Aufgabe?


4) Haben Sie schon mal an einem \"ahnlichen Experiment teilgenommen?
[ ] noch nie [ ] gelegentlich [ ] \"ofter [ ] oft [ ] sehr oft

5) Wie gut kannten Sie Ihren Experimentpartner
[ ] gar nicht [ ] fl\"uchtig [ ] ganz gut [ ] gut [ ] sehr gut

6) Konnten Sie die Aufgabe Ihren Vorstellungen entsprechend l\"osen?
[ ] ja [ ] nein [ ] mit Einschrnkungen

7) Wie bewerten Sie die Zusammenarbeit mit dem Executer?
[ ] sehr gut [ ] gut [ ] befriedigend [ ] schlecht [ ] sehr schlecht

8) Wie bewerten Sie die \"Auerungen des Executers
[ ] zu kurz [ ] ad\"aquat [ ] zu lang

9) Wie verst\"andlich waren die Beitr\"age des Executers?
[ ] sehr gut [ ] gut [ ] geht so [ ] wenig verst\"andlich
[ ] unverst\"andlich

10) Wie haben Sie sich selbst vom Executer verstanden gef\"uhlt?
[ ] sehr gut [ ] gut [ ] geht so [ ] schlecht [ ] \"uberhaupt nicht

Wenn sie Frage 10) mit "schlecht" oder "sehr schlecht" beantwortet haben:
Woran lag das?
[ ] an der Aufgabenstellung [ ] am Executer

11) Gab es Probleme, die auf das Sprachverhalten des Executers zur\"uckzufhren
waren?
[ ] ja [ ] nein

Wenn Sie Frage 11) mit "ja" beantwortet haben geben Sie bitte Beispiele!


12) Wussten Sie immer, was der Executer von Ihnen wollte?
[ ] ja, immer [ ] meistens [ ] ging so [ ] oft nicht

13) Fanden Sie das Setup geeignet f\"ur die Aufgabe?
[ ] ja [ ] nein

14) K\"onnen Sie sich vorstellen, die Aufgabe mit einem automatischen System
auszuf\"uhren?
[ ] ja [ ] nein [ ] kommt drauf an
```

```
15) Was wrden Sie in der maschinellen Aufgabe anders machen?
```

B.3.2. Questionnaires for Instruction Followers (Executors)

```
1) Wie bewerten Sie den Schwierigkeitsgrad der Aufgabe?
[ ] einfach [ ] schwerer als erwartet [ ] schwer [ ] nicht machbar

2) Wie viel Zeit haben Sie f\"ur die L\"osung der Aufgabe beansprucht?
[ ] sehr wenig [ ] weniger als erwartet [ ] mehr als erwartet
[ ] sehr viel Aufgabe war nicht l\"osbar

3) Wo lag f\"ur Sie die gr\"osste Schwierigkeit beim L\"osen der Aufgabe?


4) Haben Sie schon mal an einem \"ahnlichen Experiment teilgenommen?
[ ] noch nie [ ] gelegentlich [ ] \"ofter [ ] oft [ ] sehr oft

5) Wie gut kannten Sie Ihren Experimentpartner
[ ] gar nicht [ ] fl\"uchtig [ ] ganz gut [ ] gut [ ] sehr gut

6) Wie bewerten Sie die Zusammenarbeit mit dem Player?
[ ] sehr gut [ ] gut [ ] geht so [ ] schlecht [ ] sehr schlecht

7) Wussten Sie immer, was der Player von Ihnen wollte?
[ ] ja, immer [ ] meistens [ ] oft nicht [ ] selten

8) Wie bewerten Sie die \"Auerungen des Executers
[ ] zu kurz [ ] ad\"aquat [ ] zu lang

9) Wie verst\"andlich waren die Beitr\"age des Players?
[ ] sehr gut [ ] gut [ ] geht so [ ] wenig verst\"andlich
[ ] unverst\"andlich

10) Wie haben Sie sich selbst vom Player verstanden gef\"uhlt?
[ ] sehr gut [ ] gut [ ] geht so [ ] schlecht [ ] \"uberhaupt nicht

Wenn sie Frage 10) mit "schlecht" oder "sehr schlecht" beantwortet haben:
Woran lag das?
[ ] an der Aufgabenstellung [ ] am Executer

11) Gab es Probleme, die auf das Sprachverhalten des Players zur\"uckzufhren
waren?
[ ] ja [ ] nein

Wenn Sie Frage 11) mit "ja" beantwortet haben geben Sie bitte Beispiele!


12) Fanden Sie das Setup geeignet f\"ur die Aufgabe?
[ ] ja [ ] nein

13) K\"onnen Sie sich vorstellen, die Aufgabe mit einem automatischen System
auszuf\"uhren?
[ ] ja [ ] nein [ ] kommt drauf an
```

```
14) Was wrden Sie in der maschinellen Aufgabe anders machen?
```

References

Fernández, R., Corradini, A., Schlangen, D. & Stede, M. (2007), Towards reducing and managing uncertainty in spoken dialogue systems, *in* 'Proceedings of the 7th International Workshop on Computational Semantics (IWCS'07)', Tilburg, Netherlands.
URL: *http://www.ling.uni-potsdam.de/ das/papers/fernandez_et_al_iwcs2007.pdf*

Fernández, R., Lucht, T., Rodríguez, K. & Schlangen, D. (2006), Interaction in task-oriented human–human dialogue: The effects of different turn-taking policies, *in* 'Proceedings of the First International IEEE/ACL Workshop on Spoken Language Technology', Palm Beach, Aruba.
URL: *http://www.ling.uni-potsdam.de/ das/papers/fernandez_etal_slt2006.pdf*

Fernández, R. & Schlangen, D. (2007), Referring under restricted interactivity conditions, *in* S. Keizer, H. Bunt & T. Paek, eds, 'Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue', Antwerp, Belgium, pp. 136–139.

Fernández, R., Schlangen, D. & Lucht, T. (2007), Push-to-talk ain't always bad! comparing different interactivity settings in task-oriented dialogue, *in* 'Proceeding of DECALOG, the 11th International Workshop on the Semantics and Pragmatics of Dialogue (SemDial07)', Trento, Italy.
URL: *http://www.ling.uni-potsdam.de/ das/papers/fernandez_etal_semdial07.pdf*

Schlangen, D. & Fernández, R. (2007*a*), Beyond repair: Testing the limits of the conversational repair system, *in* S. Keizer, H. Bunt & T. Paek, eds, 'Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue', Antwerp, Belgium, pp. 51–54.

Schlangen, D. & Fernández, R. (2007*b*), Speaking through a noisy channel - experiments on inducing clarification behaviour in human-human dialogue, *in* 'Proceedings of Interspeech 2007', Antwerp, Belgium.

Siebert, A. & Schlangen, D. (2008), A simple method for resolution of definite reference in a shared visual context, *in* 'Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue', Association for Computational Linguistics, Columbus, Ohio, pp. 84–87.
URL: *http://www.aclweb.org/anthology/W/W08/W08-0113*

Siebert, A., Schlangen, D. & Fernández, R. (2007), An implemented method for distributed collection and assessment of speech data, *in* S. Keizer, H. Bunt & T. Paek, eds, 'Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue', Antwerp, Belgium, pp. 47–50.