A Multimodal In-Car Dialogue System That Tracks The Driver's Attention

Spyros Kousidis
Bielefeld University
PO-Box 10 01 31
33501 Bielefeld, Germany
spyros.kousidis@uni-bielefeld.de

Casey Kennington
CITEC, Bielefeld University
PO-Box 10 01 31
33501 Bielefeld, Germany
ckennington@cit-ec.uni-bielefeld.de

Timo Baumann
Hamburg University
Vogt-Kölln-Strasse 30
22527 Hamburg, Germany
baumann@informatik.uni-hamburg.de

Hendrik Buschmeier
CITEC, Bielefeld University
PO-Box 10 01 31
33501 Bielefeld, Germany
hbuschme@uni-bielefeld.de

Stefan Kopp
CITEC, Bielefeld University
PO-Box 10 01 31
33501 Bielefeld, Germany
skopp@uni-bielefeld.de

David Schlangen
Bielefeld University
PO-Box 10 01 31
33501 Bielefeld, Germany
david.schlangen@uni-bielefeld.de

ABSTRACT

When a passenger speaks to a driver, he or she is co-located with the driver, is generally aware of the situation, and can stop speaking to allow the driver to focus on the driving task. In-car dialogue systems ignore these important aspects, making them more distracting than even cell-phone conversations. We developed and tested a "situationally-aware" dialogue system that can interrupt its speech when a situation which requires more attention from the driver is detected, and can resume when driving conditions return to normal. Furthermore, our system allows driver-controlled resumption of interrupted speech via verbal or visual cues (head nods). Over two experiments, we found that the situationally-aware spoken dialogue system improves driving performance and attention to the speech content, while driver-controlled speech resumption does not hinder performance in either of these two tasks.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: Information Systems

Keywords

Spoken Dialogue Systems; Incremental Dialogue; In-car Dialogue; Speech Output Generation; Multimodal

1. INTRODUCTION

Vehicles are increasingly being equipped with added functionality to help the driver increase efficiency while driving, such as navigation systems and hands-free phones. However, such systems are a distraction to the driver; using the interfaces often requires some visual attention of the driver, for example, to look up a route or to find a phone number. One potential solution to this is to use spoken dialogue systems (SDS) which can understand driver commands and perform tasks. Even though this keeps the driver's visual attention on the road, it has been shown that even hands-free devices do not improve driver performance [10, 11, 12, 23]. Furthermore, simply paying attention to speech was found to induce an additional cognitive load on the driver [7]. However, according to [8], driver performance is not hindered when drivers speak to passengers, perhaps due to a shared situational awareness; the driving and traffic are, at times, topics of conversation. It was indeed found that passengers adopt strategies that relieve the driver from attending to the conversation in difficult driving situations [9]. In short, *co-location* is a requirement for risk-free in-car interaction, regardless of the interface. Most in-car systems (spoken or otherwise) do not address this, adding to the potentially already high cognitive load of the driver.

In this paper, we present our recent work on addressing this shortcoming. We have implemented a "situationally-aware" dialogue system that can react when the driving environment requires more attention from the driver. This is accomplished through *incremental* dialogue, specifically dialogue output (following [6], this term covers incremental language and speech generation here) which can interrupt its speech when a "dangerous" driving situation is detected, and flexibly resume when driving conditions become safe again. Our system delivers calendar events, informing the driver via speech about upcoming schedule items. We tested our system using a variation of a standard driving task, and found that it improved both driving performance and recall, compared to a non-adaptive baseline system. In a more recent experiment, our system yielded control of the decision to resume speaking optionally to the driver, who could signal return of attention to the spoken information, via speech or head nods. We found that this did not impact the users' driving performance or recall of information negatively. This shows that an in-car dialogue system that is situationally aware (both to extra-conversational events as well as to the dialogue) is safer and more effective than a system that has no notion of the driving conditions.

In the following section we describe the incremental and multimodal functions of our dialogue system, followed by the description of the system setup in Section 3. We then explain two experiments: one which compares performance in driving and memory

ICMI'14 November 12–16 2014, Istanbul, Turkey

ACM ACM 978-1-4503-2885-2/14/11 ...$15.00.
http://dx.doi.org/10.1145/2663204.2663244.

tasks under non-adaptive and adaptive speech conditions (Section 4); and a more recent one where we tested the effects of multimodal user control in the same tasks (Section 5). We then present and discuss our results in Section 6, and conclude in Section 7.

2. MULTIMODAL INCREMENTAL DIALOGUE

In general, (SDS) have three aspects: input via automatic speech recognition (ASR) which is the main input modality, dialogue management (DM) which makes decisions based on the input, and output in the form of natural language generation (NLG) which produces spoken utterances via speech synthesis (TTS). A multimodal SDS can handle other sources of information, such as gaze and gestures, which can be used to aid in interpreting the intent of the speaker as in [15], and help the DM make more informed decisions on how to act. In this paper, we use a multimodal SDS by incorporating information about the driving situation, ASR, and head nods into the DM decision.

An SDS that processes input *incrementally* produces behaviour that is perceived by human users to be more natural than systems that end-point on larger sentence-length segments or use a turn-based approach (such as phone-based dialogue systems) [1, 25, 24]. Incremental dialogue has seen improvements in speech recognition where intermediate results are provided before an utterance is complete [2], and speech synthesis where parts of an utterance yet to be realised can be altered, even while the utterance prefix is in progress [6]. This approach to interactive dialogue is necessary in real-time interactive systems, such as in-car dialogue, as the dialogue system can act and react immediately to changes in the driving situation.

Our SDS consists of many incremental components and is able to integrate many multimodal features. In the following, we describe the most relevant components used in this study, namely incremental language generation and head nod detection.

Incremental Output Generation.

Making the output of an in-car SDS situationally aware requires its output generation modules, speech synthesis and natural language generation, to be able to (1) timely and plausibly interrupt and resume speech output, and (2) to flexibly adapt or even reformulate the content of its utterances, taking into account a preceding delivery interruption.

Both requirements call for incremental processing in these modules. On the level of speech synthesis, incrementality allows for shorter response times (i.e., it can resume faster) as the system can start speech output while still synthesising the rest of an utterance [6]. It also enables changes to the prosody of an ongoing utterance [4], allowing the system to add a prosodic marker to signal the system's awareness to the word preceding the interruption. On the level of natural language generation, incrementality makes it possible to change those parts of an utterance that have not been delivered yet. The continuation of an interrupted utterance can thus differ from planned but yet undelivered parts by choosing a continuation point that, e.g., re-states some of the context but does not repeat more than is needed.

Our work builds on the existing incremental output generation system of [6] that fulfils the requirements specified above and is partially available in the open source incremental dialogue processing toolkit INPROTK ([5], see below)[1]. It consists of incremental components for speech synthesis and natural language generation that are integrated in such a way that timely interruptions and adaptive continuations are possible.

The system's language generation component creates utterances in two processes [6]. The first process plans the overall utterance by laying out a sequence of chunks which determine what will be said, and when; the second, which is based on the SPUD microplanning framework [26], computes how each of these chunks is realised linguistically. Utterances are incrementally generated chunk by chunk. Adaptations to an ongoing utterance are therefore constrained to the chunk level as well. The chunk-planning process can change the sequence of chunks, repeat one or several chunks, or leave some out. The microplanning process can change how a chunk is realised, e.g., by inserting or leaving out cue words, by providing information that has been mentioned before, or by making information conveyed implicitly explicit – or vice versa. Our system made use of adaptations resulting from both processes.

Incremental speech synthesis [4] performs all computationally expensive processing steps, such as waveform synthesis, as late as possible while performing prosodic processing (which has non-local effects) as early as necessary [3], resulting in fast response times with as little impact on quality as possible. Ongoing synthesis can be changed, and adapted prosodically with minimal latency, and provides detailed progress information on various linguistic levels. Our system uses the incremental capabilities to stop synthesis at word boundaries when interrupted and to generate new sentence onset intonations for continuations.

Multimodal Dialogue with Head Nods.

A new addition to our SDS is the incorporation of head nods as an additional modality. Head nods in human–human interaction are the most prominent head gesture, especially in situations of active listening [27]. In dyadic interaction, head nods signal understanding/agreement and prompt the speaker to continue, among other functions [22]. These properties suggest the head nod gesture as a good alternative to speech for a natural prompt from the driver to the in-car system.

Our head nod detection component uses the output of a head tracking software (described below) that utilises a standard webcam. The software estimates the head posture from the position and orientation of the face mask shown in Figure 1. Head posture is defined by the standard translation (x, y, z) and rotation $(pitch, yaw, roll)$ axes. In addition to the head posture, the head tracking software reports a percentage value representing the tracking confidence.

Figure 1: Face capture frame from the face tracking software FaceAPI.

Algorithm 1 shows our simple algorithm that detects head nods incrementally based on the raw pitch rotation angle (see Figure 2) and the tracking confidence. We calculate the energy of the head pitch rotation at each sample (30 Hz), and open a new "window" (candidate of a head nod) if the energy and confidence both exceed their pre-set thresholds. If enough high-energy points extend the window beyond a *minimum duration*, the algorithm reports an (ongoing) head nod. If no new high energy points extend the window any more, the algorithm reports that the head nod ceased. This al-

[1] `http://inprotk.sourceforge.net`

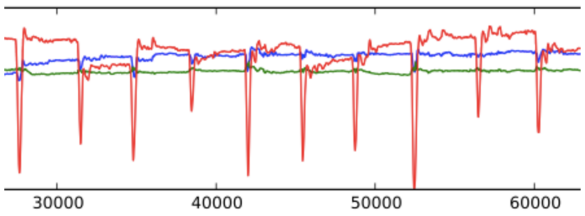


Figure 2: Head rotation angles as detected by our real-time head nod detection software. The pitch rotation angle spikes (red) signify instances of head nods.

gorithm is rather simple and would most probably be unsuitable for spontaneous human–human dialogue, where the form of head nods is much more varied [18]. However, we qualitatively evaluated that the algorithm worked well for head nods that are presented to the system, as they are *displayed* by the participants, rather than being spontaneous. The parameters of the algorithm were fine-tuned for each participant before experiments.

3. MULTIMODAL SYSTEM SETUP

Hardware.

Our driving simulation scenario consists of a 40-inch 16:9 screen with a Thrustmaster PC Racing Wheels Ferrari GT Experience steering wheel and pedal. Audio is presented to the participant via Sennheiser headphones (see Figure 3). To recognise speech, we used a Sennheiser ME 2 lavalier microphone and for head nod detection, we used a Logitech C270 webcam.

Figure 3: View of experiment setup: a driving simulation, interfaced with a steering wheel and pedal, is shown on the large screen. Speech is presented to the participant via headphones

Software.

For the driving simulator, we used the OpenDS Toolkit [21][2]. We developed our own simple driving scenarios (derived from the "ReactionTest" task, distributed together with OpenDS) that specified the driving task and timing of the concurrent speech, as described below. For head tracking, we used SeeingMachines FaceAPI[3] which sent head posture data to our head nod detection algorithm described above. Our incremental SDS was built on the framework of INPROTK [5] (our speech output – NLG and TTS – componens are part of this), with recent extensions allowing for multimodal coupling of distributed components which we denote INPROTK$_S$ [16]. We make use of an incremental automatic speech recogniser (a variant of Sphinx as part of INPROTK [2]) to recognise specific words. For DM, we integrated OpenDial [20][4] into INPROTK$_S$.

Algorithm 1 Head Nod Detection: the DETECT_HEAD_NOD procedure is called every time new data is received from FaceAPI. The four variables that are used for thresholds and housekeeping are described before the procedure.

```
confidence_threshold  # head confidence threshold (%)
velocity_sample  # max time jump for calculating velocities (ms)
min_duration  # min duration for buffer before detecting a nod (ms)
buffer  # energy buffer for time jump (samples)
 1: procedure DETECT_HEAD_NOD(frame)
 2:     if frame.confidence < confidence_threshold:
 3:         return
 4:     end if
 5:     if frame.time – lastframe.time < velocity_sample:
 6:         velocity =frame.head_rotation – lastframe.head_rotation
 7:         energy = 100∗velocity^2
 8:          # (scale to deal with small numbers)
 9:     else energy = 0
10:     end if
11:     if energy > energy_threshold:
12:         buffer.append(frame)
13:     end if
14:     if buffer.length > min_duration:
15:         send_headnod_event()  # (head nod detected)
16:     end if
17:     lastframe = frame
18: end procedure
```

Connecting All Components with mint.tools.

Our overall system setup is depicted in Figure 4. The three hardware components (and their software), which comprise three modalities of driving scenario events, speech, and head nods, were plugged into three corresponding workstations (all used Ubuntu 12.04, except the one running the FaceAPI software, which run on MS Windows 7). An important aspect of our setup is how the various software components communicate with each other across the network. For this, we used the *mint.tools* architecture [19], which utilises the Robotics Service Bus (RSB) message passing architecture [28] and the InstantIO/InstantReality framework, facilitating real-time data passing and logging[5,6].

FaceAPI sent raw data (via InstantIO) to a process running our implemented algorithm that detected head nods, which in turn passed head nod events via RSB to INPROTK$_S$. To make our system situationally aware, we modified OpenDS to pass real-time data (e.g. car position/velocity/events in the simulation, such as a gate becoming visible or a lane change) to INPROTK$_S$ (via InstantIO). Finally, ASR and NLG were controlled by INPROTK$_S$ directly. Further information was sent from INPROTK$_S$ (e.g., input events such as head nods or detected speech, DM decisions, text used for NLG) to the logger (via InstantIO). Running on a dedicated fourth workstation, the logger wrote all data (with timestamps) sent over InstantIO from any of the components to a compressed file in XML format, which we used for post analysis. We want to further add that with *mint.tools*, one can use the XML log file to replay the recorded interaction in real-time.

In the following we describe two experiments. Experiment 1 was performed in previous work presented in [14], but as the setup is a precursor, and the results are directly comparable to the work done

[2] http://www.opends.eu/
[3] http://www.seeingmachines.com/product/faceapi/
[4] http://opendial.googlecode.com/
[5] https://code.cor-lab.de/projects/rsb
[6] http://www.instantreality.org/

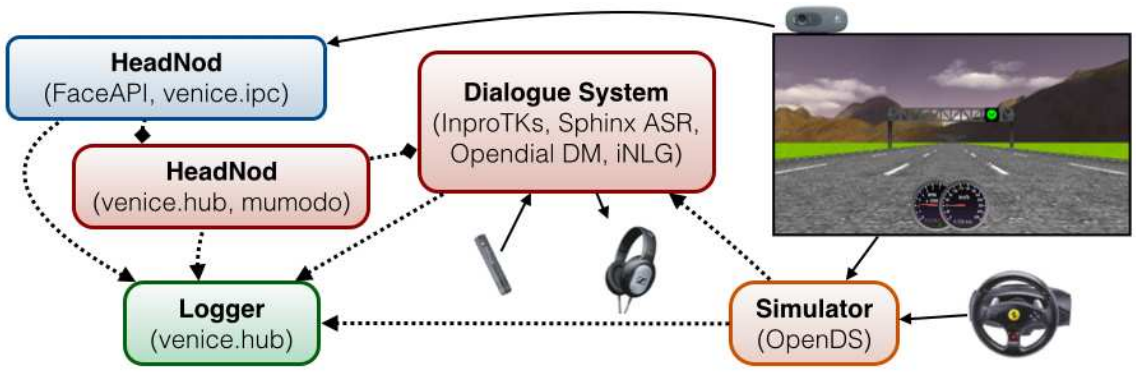


Figure 4: Overview of our system: solid lines denote connections between hardware and software components, colours denote the workstations (in this case 4) that we used. Dashed lines denote network connections between components, arrows denote InstantIO connections, diamonds denote RSB connections.

in this paper, we also provide a description here. Experiment 2 builds on Experiment 1, with added control over the SDS given to the driver.

4. EXPERIMENT 1: ESTABLISHING ADAPTIVE SPEECH

The goal of this experiment is two-fold: first, we want participants to be able to perform a driving task as a responsible driver would; second, we want to explore how well they pay attention to and recall speech during driving, under two possible presentations of speech: the *adaptive* presentation, in which the speech of the SDS is interrupted when a "dangerous" situation is detected in the driving scene, and later resumed after the dangerous situation is no longer present. This mimics a situated dialogue participant who is aware of the physical surroundings and driving conditions; and the *non-adaptive* presentation, a non-incremental system that does not stop speaking when a dangerous driving condition is detected. In order to simulate these conditions, we use a combination of two tasks: a driving task and a memory task, which we explain in detail below.

The Driving Task.

For the driving task we used a variant of the standard lane-change task (LCT [13]). It requires the driver to react to a green light positioned on a signal gate above the road (see Figure 5). The driver, otherwise instructed to remain in the middle lane of a straight, 5-lane road, must move to the lane indicated by the green light, remain there until a tone is sounded, and then return again to the middle lane. OpenDS gives a *success* or *fail* result to this task depending on whether the target lane was reached within 10 seconds (if at all) and the car was in the middle lane when the signal became visible. In addition, OpenDS reports a *reaction time*, which is the time between the moment the signal to change lane becomes visible and the moment the lane has been reached. A lane-change trial simulates a "dangerous" situation on the road.

We added a second component to the task, which was to change the speed from 40 km/h (the default speed that the car maintained without the gas pedal being pressed) to 60 km/h during the lane change. This speed is lower than the maximum speed, so that the right position of the gas pedal had to be found and the speed be monitored continuously.

The Memory Task.

We tested the attention of the drivers to the generated speech using a simple true/false memory task. The dialogue system generated calendar-entry utterances such as *"am Samstag den siebzehnten Mai 12 Uhr 15 bis 14 Uhr 15 hast du 'gemeinsam Essen im Westend mit Martin'"* (on Saturday the 17th of May from 12:15-14:15 you are meeting Martin for lunch). These utterances (spoken by a female voice) always had 5 information tokens (chosen at random from a database) in a specified order: day, time, activity, location, and partner. Three seconds after the utterance was complete, and while no driving distraction occurred, a true/false confirmation question about one of the uttered tokens was asked by a male voice, e.g. *"Richtig oder falsch? – Freitag"* (Right or wrong? – Friday). The subject was then required to answer true or false by pressing one of two respective (labelled) buttons on the steering wheel. The token of the confirmation question was chosen randomly.

Figure 5: From the perspective of the driver, a gate is shown with a green lane signal over the right-most lane.

In the case of an interruption/resumption, tokens spoken after the resumption can be more easily remembered than those given before the interruption. By giving the early tokens (day and time) a higher probability of occurrence, we biased the design *against* the adaptive system since the question tends to refer to tokens spoken *before* the interruption more often than not.

Interaction Between Tasks.

Figure 6 shows how the task unfolds over time when changing the lane: all red-dashed lines represent pre-set event triggers (that are invisible to the driver) or simply events of the simulation that trigger unique messages to be sent to the SDS. At the t_1 marker, a trigger is sent to the DM to start speaking. A random delay (0–4 seconds for the non-adaptive, 4–7 seconds for the adaptive setting) is inserted before the speech begins in order to vary the type of token that is spoken during the exact moment of interruption or steering. At t_2, the gate is in view (as seen from Figure 5) and a gate light is visible. In the adaptive setting, at this point the speech would be

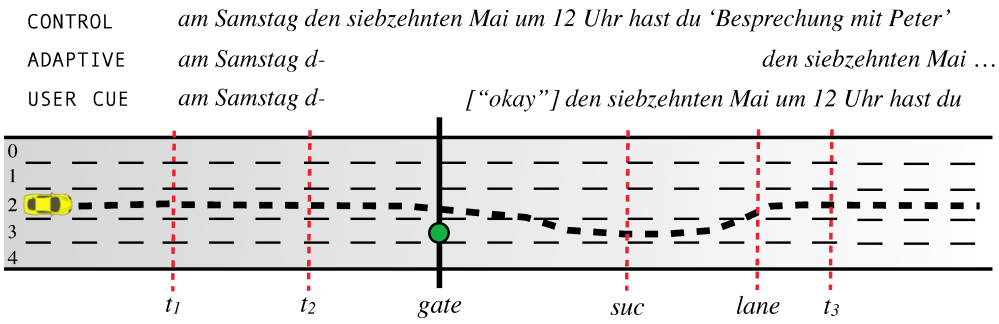


Figure 6: Top view of driving task: as the car moves to the right over time, speech begins at t_1, the gate with the lane-change indicator becomes visible at t_2, where in the adaptive version speech pauses. If a successful lane change is detected at *suc*, then speech resumes at *lane*, otherwise it resumes at t_3. All red-dotted lines denote events sent from OpenDS to the Dialogue Manager. CONTROL and ADAPTIVE were used in Experiment 1, whereas ADAPTIVE and USER CUE were used in Experiment 2. In USER CUE, the participant speaks "okay", triggering a continuation in the speech before the end of the lane-change.

interrupted; in the non-adaptive setting the speech would continue until complete. At *suc*, the target lane has been reached (the tone signal is sounded), but the speech does not resume yet. At *lane*, the car has returned to the middle lane, at which point the adaptive speech would continue the interrupted utterance. In case the task was not completed correctly (e.g. the target lane change did not happen), a time-out at t_3 would trigger the adaptive speech to continue. Three seconds after the calendar event speech is completed, the true/false question is asked. There is ample time for the participant to press the yes/no button before any other distraction (on the road or from speech) occurs.

A single driving scenario consisted of 44 gates, with 11 gates each for the following conditions: speech with no lane change, lane change with no speech, non-adaptive speech with lane change, and adaptive speech with lane change. Each of these gates was followed by two empty gates, in order to clearly separate the effects of successive trials. The order of the gate types was randomly shuffled and unique for each participant. This, combined with the random silence before speech, makes the events (speech or lane-change) to be perceived as occurring completely randomly.

Procedure.

Figure 3 shows a participant during the driving simulation experiment. First, each participant signed a consent form and was then seated in the chair in front of the steering wheel and screen (seat adjustments made, if necessary). The participant was then given headphones to put on, after which the audio levels were tested and the task was explained. Following this, the OpenDS scene was started, showing the driving simulation on the large screen, at which point the participant was instructed to control the steering wheel and pedal. In the beginning of the simulation, 10 signal gates were presented for practice in using the controls and performing the driving task. During this practice stretch of road an experimenter was sitting next to the participant in order to clarify any questions that could be asked during this phase (the simulation could be paused, if necessary, to answer difficult questions or make adjustments). When the participant confirmed that he or she had understood the task, the experimenter left the scene.

Immediately after the practice gates, without any interruption, a clearly marked "START" gate signalled the beginning of the experiment, followed by the sequence of 44 gates described previously. The end of the experiment was signalled with a clearly marked "FINISH" gate, at which point the simulation stopped. In total, the driving simulation took around 30 minutes, including practice time. The participant was then given a post-task questionnaire to fill out.

In total, 17 participants (8 male, 9 female, aged 19–36) participated in the study. All of the participants were native German speakers affiliated with Bielefeld University and holders of a driving license (for normal automobiles). Two participants had previous experience with driving simulators and only one had previous experience with SDS.

5. EXPERIMENT 2: ADAPTIVE SPEECH WITH DRIVER CONTROL

For this experiment, the driving task and the memory task were the same as in Experiment 1, but the interaction between them and the presentation of the gates were altered. This difference is explained in the following.

Interaction Between Tasks: Added Driver Control.

In this experiment, participants were presented with two different systems. *System A* interrupted speech at the beginning of a lane-change task and continued after the task was complete, as described before. *System B* had the added functionality of allowing the participant to cue the SDS to continue speaking after its speech had been interrupted. The cue could be given in three ways, either by saying "okay" or by saying "weiter" (continue), which were the only two words we allowed our SDS to react to. The participant could also perform a simple head nod. Cues from the driver were only allowed while the calendar event was interrupted (i.e., the driver could not cue the system to begin a new calendar event). The participant could also make no cue at all, allowing the system to function normally (as in System A) by resuming the speech after the lane change was completed. This simple alteration gives the driver the option of controlling when the continuation of the calendar event speech is started. Figure 6 gives an example of how giving a cue ("okay") could play out over time during a lane-change task.

Procedure.

In this experiment, participants were seated, audio levels were tested, head-nod detection parameters were fine-tuned, and finally the task was explained. As in the previous experiment, 10 training gates without calendar events were used to familiarise the participants to the driving task. Another 4 gates with calendar events were added in order to familiarise participants with controlling the speech resumption by means of speech and head nod cues. Each participant had to complete one session of System A and one of System B (20 gates each). Half of the participants were presented with System A first, then System B, the other half in the opposite or-

der. When System B was presented first, participants were presented with 10 + 4 training gates as described above, followed by the main task ("START" , 20 task gates, "FINISH"). Immediately afterwards, OpenDS was restarted and participants were presented with System A ('START", 20 gates, "FINISH"). When system A was presented first, the ASR and head nod tests, as well as the 4 training gates occurred after System A was completed and immediately before System B was started. Figure 7 shows the system presentations graphically.

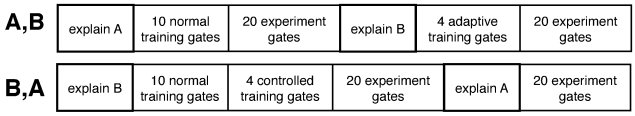


Figure 7: Experiment 2 procedure for varied system presentation. System B responded to driver cues, System A did not.

In this second study 10 native German speakers (3 female and 7 male) participated, two of whom had no drivers license, half had previous experience with SDS and half had previous experience with driving simulators. One participant was familiar with the specific system, having participated in Experiment 1.

6. RESULTS

We present here the results from both experiments for comparison. The results from Experiment 1 have been previously reported in [17] and in more detail in [14].

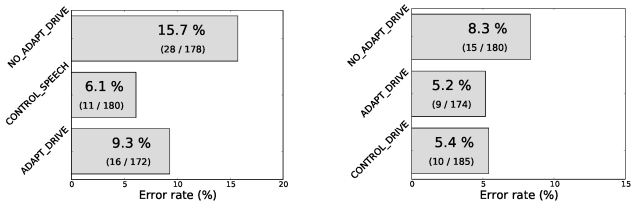
6.1 Experiment 1

In our first experiment, we found that listening to speech from a system that is not co-located adversely affects the performance in both the driving task and the memory task. In contrast, a system that is adaptive and aware of the driving conditions leads to participants' performance that is equivalent to the control conditions of performing either task in isolation. The *error rate* quantifies the percentage of unsuccessful trials in the driving (e.g., target lane not reached in time) and memory task (e.g., wrong or no answer given; see Figure 8). Statistical significance was tested using a Generalised Linear Mixture Model (GLMM) for estimating the probability of a successful/unsuccessful response as a function of within-subject factors CONDITION and TIME as well as many between subject factors (e.g., AGE, GENDER). In both the driving and memory tasks the effect of CONDITION was found to be significant ($p < 0.05$) while no significant effect was found for any of the between-subject factors. These findings validated our hypotheses and are consistent with previous research.

Interestingly, the majority of participants in the first experiment stated (in the post-experiment questionnaire) their preference towards the non-adaptive system, as they considered the system's interruption to be a hindrance. Some stated that they would like to be able to control the system in some way (giving rise to Experiment 2). There were further results in Experiment 1 that are not of interest here and for which we refer the reader to [14].

6.2 Experiment 2

In the second experiment, we provided the participants with the option to trigger resumption of the output using either speech or head nods. We wanted to test whether this type of control would affect performance, as well as how it would be adopted by the participants. We find almost identical performance in both systems in the memory task, and no significant difference in the driving task (see Figure 9). Significance was tested with the same method (GLMM) as in Experiment 1.



(a) Memory task (b) Driving task

Figure 8: Error rates in the two tasks in Experiment 1. For both tasks, there were significant differences between the adaptive and non-adaptive conditions, with no significance between adaptive and control conditions.

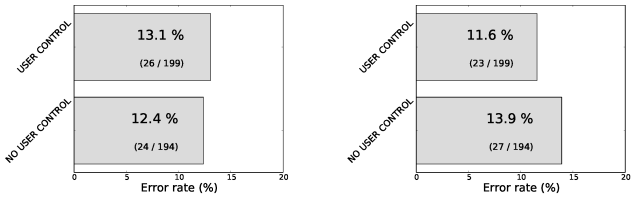
Therefore, we find no evidence that our introduction of control of the system speech resumption degrades performance compared to a system that decides by itself. We note that participants made wide use of the functionality (see below), asking the system to resume when they felt comfortable with their tasks or concentrating on the driving task more when they needed to.

As expected, the vast majority (9 out of 10) of the participants stated their preference towards the system that allows user control, and specifically by speech rather than head nods. The latter finding is implicit, as head nods were used less than speech when both options were available (the detection of head nods and the automatic speech recognition performed roughly equally well). Noting that half of the participants had previous experience with SDS, we could interpret this as an effect of collective familiarity with SDS that is just not there for "head gesture interfaces" yet; that is, the general public may not be as accustomed to systems understanding gestures, as opposed to systems understanding speech, at least in some contexts such as in-car systems. Since our findings from Experiment 2 are novel, we present them in more detail below.

Effect of User Control.

We present here results of a preliminary analysis of the effect of *usage* of user control on the performance in both task. As the usage was optional, we firstly look at amount of usage, followed by the error rate in both tasks, as shown in Table 1. Note that SPEECH and HEAD NOD only refer to correctly detected events that actually triggered the resumption. The low scores in the TIME-OUT condition represent an aggregation of errors and added cognitive load due to mis-recognitions. We have found no evidence that user-guided resumption of speech can lead to decreased performance in either task. As mentioned in the previous section, speech was the preferred option compared to head nods.

We further investigated the effect of the *time* of the system speech resumption on the main variables, regardless of whether the participant controlled the resumption or not (Figure 10). For both tasks, there appears to be an optimal time for the speech to resume, or conversely, two inappropriate times. Recall that the system speech

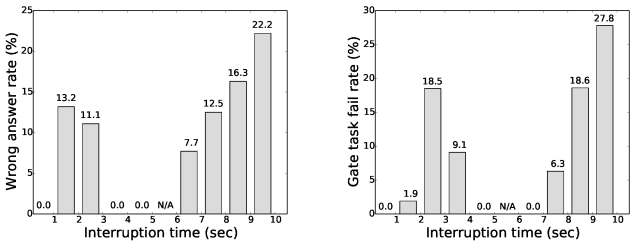


(a) Memory task (b) Driving task

Figure 9: Error rates in the two tasks in Experiment 2.

Table 1: Error rate in two tasks by type of evens triggering resumption of system speech.

		Error rate (%)	
Resume by	Usage (%)	Driving task	Memory task
Speech	46.7	7.5	11.8
Head Nod	9.0	5.5	5.5
Return to lane	25.1	2	14
Time-out	18.6	35	18.1



(a) Memory task (b) Driving task

Figure 10: Error rates in the two tasks by duration of system speech interruption.

interrupts as soon as the signal indicating a lane change becomes visible. An early "peak" appears around 2-3 seconds, which coincides roughly with the main action of the driving task, which is to accelerate and steer in order to reach the target lane. After this time, the driving task becomes much easier: it is relatively easy to maintain 60 km/h after first stabilizing the car velocity at that level and there is nothing to do other than wait for the tone signal to return to the middle lane. This is reflected by the valley in the middle region in both tasks, during which no errors occur. The error rates begin to scale up again after 6 seconds, where again several effects and factors aggregate: second or third attempts after mis-recognitions of speech or nods, longer mid-utterance silence time that affects recall, and resumptions after a failure in the driving task has been registered.

It would be possible to further refine the system behaviour to avoid re-starting the speech at sub-optimal times. This can be implicitly coded by monitoring the steering and acceleration, in order to estimate the cognitive load of the driving task in real time. Such estimation (which we defer to future work) would also be a more realistic approach to triggering speech interruptions as well. Here we have assumed that the system "knows" when to stop speaking.

Individual Differences.

Although the overall performance in the memory task is roughly equal for the two systems, we see in Figure 11 that individual participants' performances vary in the direction of the effect. However, these differences are for the most part relatively small; they amount to 1–2 gates per participant/system.

In the case of the driving task, we also notice (see Figure 12) that individual participants can be affected in either direction or not at all. Interestingly, participants who do not perform well at the driving task do not coincide with those that have no driver's license or experience with driving simulators. Further, performance does not correlate across the two tasks (i.e., individual participants may perform well in either, neither, or both tasks). Participants in general put more effort and attention on the driving task, which is the desired behaviour. Giving drivers control over the speech delivery did not hinder their driving performance.

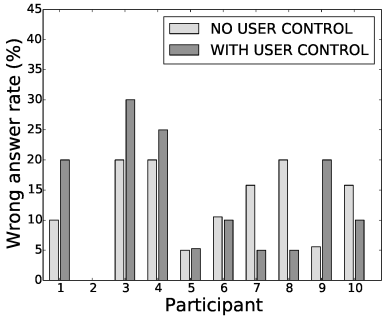


Figure 11: Effect of system type (with/without user control) on individual participant's performance in the memory task

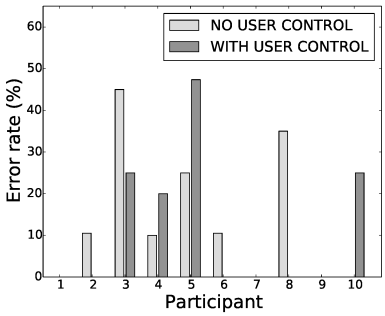


Figure 12: Effect of system type (with/without user control) on individual participant's performace in the driving task

7. CONCLUSIONS

We have presented the latest developments of our co-located, situationally-aware multimodal in-car dialogue system. Our explorations so far have shown that situational awareness is indeed a significant property for in-car systems that are safe, as well as more efficient in their own goal (as represented by the memory task). Further, we have added the functionality of yielding control to the driver using natural human interaction modalities (speech and head gestures). In our experiments, we did not find any evidence that this by itself has any effect on performance, but further work is required in this direction. We found that participants were much more comfortable speaking than nodding, and were forgiving of mis-recognitions in speech more than in head nods. In future work, we will further investigate the best division of labour between such modalities.

8. ACKNOWLEDGEMENTS

This research was partly supported by the Deutsche Forschungsgemeinschaft (DFG) in the CRC 673 "Alignment in Communication", the Center of Excellence in "Cognitive Interaction Technology" (CITEC), and a PostDoc grant by Daimler and Benz Foundation to Timo Baumann. The authors would like to thank Oliver Eickmeyer and Michael Bartholdt for helping implement the system setup, as well as Gerdis Anderson and Fabian Wohlgemuth for assisting as experimenters. Thanks also to the anonymous reviewers.

9. REFERENCES

[1] G. Aist, J. Allen, E. Campana, L. Galescu, C. A. G. Gallo, S. Stoness, M. Swift, and M. Tanenhaus. Software architectures for incremental understanding of human speech. In *Proc. Interspeech*, pages 1922–1925, 2006.

[2] T. Baumann, M. Atterer, and D. Schlangen. Assessing and Improving the Performance of Speech Recognition for

Incremental Systems. In *Proc. NAACL-HLT*, pages 380–388, 2009.

[3] T. Baumann and D. Schlangen. Evaluating prosodic processing for incremental speech synthesis. In *Proc. Interspeech*, pages 438–441, 2012.

[4] T. Baumann and D. Schlangen. Inpro_iss: A component for just-in-time incremental speech synthesis. In *Proc. ACL 2012 System Demonstrations*, pages 103–108, 2012.

[5] T. Baumann and D. Schlangen. The InproTK 2012 release. In *NAACL-HLT Workshop SDCTD*, pages 29–32, 2012.

[6] H. Buschmeier, T. Baumann, B. Dosch, S. Kopp, and D. Schlangen. Combining incremental language generation and incremental speech synthesis for adaptive information presentation. In *Proc. SIGDIAL2012*, pages 295–303, 2012.

[7] V. Demberg, A. Sayeed, A. Mahr, and C. Müller. Measuring linguistically-induced cognitive load during driving using the ConTRe task. In *Proc. Automotive'UI*, pages 176–183, 2013.

[8] F. A. Drews, M. Pasupathi, and D. L. Strayer. Passenger and cell-phone conversations in simulated driving. *Proc. HFES 48th Annual Meeting*, pages 2210–2212, 2004.

[9] F. A. Drews, M. Pasupathi, and D. L. Strayer. Passenger and cell phone conversations in simulated driving. *Journal of Experimental Psychology: Applied*, 14(4):392–400, 2008.

[10] J. He, A. Chaparro, B. Nguyen, R. Burge, J. Crandall, B. Chaparro, R. Ni, and S. Cao. Texting while driving: Is speech-based texting less risky than handheld texting? In *Proc. Automotive'UI 13*, pages 124–130, 2013.

[11] W. J. Horrey and C. D. Wickens. Examining the impact of cell phone conversations on driving using meta-analytic techniques. *Human Factors*, 48:196–205, 2006.

[12] Y. Ishigami and R. M. Klein. Is a hands-free phone safer than a handheld phone? *Journal of Safety Research*, 40(2):157–164, 2009.

[13] ISO. Road vehicles – Ergonomic aspects of transport information and control systems – Simulated lane change test to assess in-vehicle secondary task demand. ISO 26022:2010, 2010.

[14] C. Kennington, S. Kousidis, T. Baumann, H. Buschmeier, S. Kopp, and D. Schlangen. Better driving and recall when in-car information presentation uses situationally-aware incremental speech output generation. In *Proc. AutoUI*, 2014.

[15] C. Kennington, S. Kousidis, and D. Schlangen. Interpreting situated dialogue utterances: An update model that uses speech, gaze, and gesture information. In *Proc. SIGDIAL2013*, pages 173–182, 2013.

[16] C. Kennington, S. Kousidis, and D. Schlangen. InproTKS: A toolkit for incremental situated processing. *Proc. SIGDIAL2014: Short Papers*, pages 84–88, 2014.

[17] S. Kousidis, C. Kennington, T. Baumann, H. Buschmeier, S. Kopp, and D. Schlangen. Situationally aware in-car information presentation using incremental speech generation: Safer, and more effective. In *Proc. EACL 2014 Workshop on Dialogue in Motion*, pages 68–72, 2014.

[18] S. Kousidis, Z. Malisz, P. Wagner, and D. Schlangen. Exploring annotation of head gesture forms in spontaneous human interaction. In *Proc. TiGeR*, 2013.

[19] S. Kousidis, T. Pfeiffer, and D. Schlangen. MINT.tools: Tools and adaptors supporting acquisition, annotation and analysis of multimodal corpora. In *Proc. Interspeech 2013*, pages 2649–2653, 2013.

[20] P. Lison. Probabilistic dialogue models with prior domain knowledge. In *Proc. SIGDIAL2012*, pages 179–188, 2012.

[21] R. Math, A. Mahr, M. M. Moniri, and C. Müller. OpenDS: A new open-source driving simulator for research. In *GMM-Fachbericht-AmE 2013*, 2013.

[22] E. Z. McClave. Linguistic functions of head movements in the context of speech. *Journal of Pragmatics*, 32(7):855–878, 2000.

[23] S. P. McEvoy, M. R. Stevenson, A. T. McCartt, M. Woodward, C. Haworth, P. Palamara, and R. Cercarelli. Role of mobile phones in motor vehicle crashes resulting in hospital attendance: A case-crossover study. *BMJ*, 331(7514):428, 2005.

[24] G. Skantze and A. Hjalmarsson. Towards incremental speech generation in dialogue systems. In *Proc. SIGDIAL2010*, pages 1–8, 2010.

[25] G. Skantze and D. Schlangen. Incremental dialogue processing in a micro-domain. In *Proc. EACL*, pages 745–753, 2009.

[26] M. Stone, C. Doran, B. Webber, T. Bleam, and M. Palmer. Microplanning with communicative intentions: The SPUD system. *Computational Intelligence*, 19:311–381, 2003.

[27] P. Wagner, Z. Malisz, and S. Kopp. Gesture and speech in interaction: An overview. *Speech Communication*, 57:209–232, 2014.

[28] J. Wienke and S. Wrede. A middleware for collaborative research in experimental robotics. In *Proc. 2011 IEEE/SICE SII2011*, pages 1183–1190, 2011