Analyzing the Effects of Spoken Dialog Systems on Driving Behavior

Jeongwoo Ko[1], Fumihiko Murase[1], Teruko Mitamura[1], Eric Nyberg[1], Masahiko Tateishi[2], Ichiro Akahori[2]

[1]Language Technologies Institute, School of Computer Science, Carnegie Mellon University
5000 Forbes Ave. Pittsburgh PA 15213, USA
{jko, fmurase, teruko, ehn}@cs.cmu.edu
[2]Research Laboratories, DENSO CORPORATION
500-1, Minamiyama, Komenoki-cho, Nisshin-shi, Aichi-ken, Japan
mtatei@rlab.denso.co.jp, iakahori@its.denso.co.jp

Abstract

This paper presents an evaluation of a spoken dialog system for automotive environments. Our overall goal was to measure the impact of user-system interaction on the user's driving performance, and to determine whether adding context-awareness to the dialog system might reduce the degree of user distraction during driving. To address this issue, we incorporated context-awareness into a spoken dialog system, and implemented three system features using user context, network context and dialog context. A series of experiments were conducted under three different configurations: driving without a dialog system, driving while using a context-aware dialog system, and driving while using a context-unaware dialog system. We measured the differences between the three configurations by comparing the average car speed, the frequency of speed changes and the angle between the car's direction and the centerline on the road. These results indicate that context-awareness could reduce the degree of user distraction when using a dialog system during driving.

1. Introduction

Dialog systems for automotive environments face many new challenges which do not exist for traditional spoken dialog systems. Such challenges include robust task management during network loss and intelligent user assistance in new task environments. To handle these challenges, the dialog system must sense important changes in the user's environment and communicate them to the user. In the field of human-computer interaction, context-awareness (Schilit et al, 1994; Dey and Abowd, 2000) has been developed as a means to offer tailored information to users in dynamically changing environments. We incorporated context-awareness into a spoken dialog system, and implemented three system features using user context (e.g. user preferences, location), network context and dialog context.

The user context is used for location-aware suggestions which provide information that the driver might find valuable given the current context. For example, the system may suggest nearby rest areas when the driver has been driving for a long time, or notify him/her of the unavailability of a destination. As network connectivity can be unpredictable in automotive environments, management of the network context is important. When the wireless network signal is lost, the user can still use the dialog system in a reduced capacity (e.g. to control in-car devices). When the network connection is re-established, the system downloads information for any pending requests, and automatically restarts any pending task dialogs. The system will purge pending tasks that are assumed to be no longer relevant, depending on the change in time and location since the task was interrupted. As dialogs using mobile and/or pervasive devices tend to be more event-driven, and involve more task-switching (McTear, 2004), the system incorporates the dialog context to support smooth dialog task switching and robust reference resolution.

To evaluate context-awareness in a spoken dialog system, we compared three different configurations: a) driving without using a dialog system (baseline), b) driving while using a dialog system which supports context-aware features (context-aware), and c) driving while using a dialog system which does not support context-aware features (context-unaware). We measured the differences between the three configurations by comparing the average car speed, the frequency of speed changes and the angle between the car's direction and the centerline on the road. Analysis of speed shows that the variance in car speed was greater in the context-unaware case than in the baseline case. In addition, participants changed car speed more frequently when using the context-unaware system than when using the context-aware system. Further analysis on the angle between the car's direction and the centerline shows that less change was noted when using the context-aware system. These results indicate that context-awareness could reduce the degree of user distraction when using a dialog system during driving.

The remainder of this paper is structured as follows: Section 2 describes the CAMMIA dialog system which supports the context-aware features described above. Section 3 explains the experimental design, and Section 4 presents the experimental results. Finally, Section 5 concludes with suggestions for future research.

2. CAMMIA Dialog System

The CAMMIA (Conversational Agent for Multilingual Mobile Information Access) system is a context-aware multilingual spoken dialog system which provides route guidance and information services in English and Japanese (Nyberg et al., 2002; Obuchi et al., 2004; Ko et al., 2006a). CAMMIA supports multimodality in the form of a speech interface combined with a tactile screen.

CAMMIA consists of four layers to support multi-modal, mobile information access: the user interface layer for low-level user interaction, the dialog management layer to support context-aware features, the task management layer for data download management and the application layer which includes databases and wrappers.

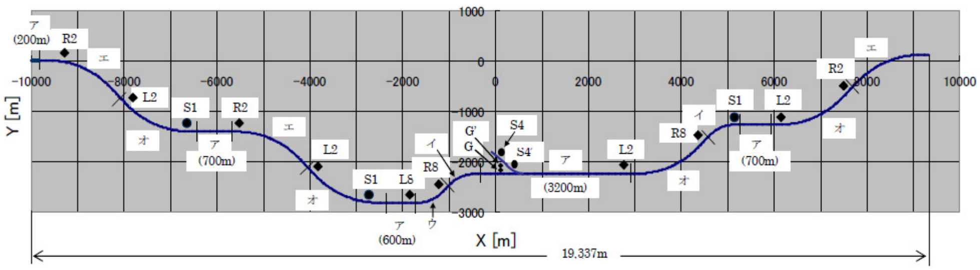


Figure 2. Driving course in driving simulator.

The dialog management layer is important for context-aware services. It consists of two subcomponents: the Dialog Manager and the Reasoner. The Dialog Manager represents and manages multiple ongoing topics of conversation (Nyberg et al., 2002). Dialogs are represented as dialog scenarios: each scenario consists of a set of states and transitions between states. To support multiple dialog topics, the attentional stack model (Allen, 1995) has been incorporated. Figure 1 shows an example of flexible dialog switching based on the stack model. When the user starts a new dialog topic (U1), the system creates a new sightseeing dialog in the stack. For the utterance U4, the system pushes a new dialog into the stack and extracts the slots from the previous dialog in stack. Therefore, instead of asking location and time which are necessary slots for the weather dialog, the system directly answers user's question. To handle utterance U5, the system automatically goes back to the sightseeing dialog from the weather dialog because the weather dialog does not support price information. As the current state in the sightseeing dialog includes information about Echo Valley, the system then provides price information for Echo Valley.

U1: I am looking for a ski resort in Nagano.
S1: There are two famous ski resorts in Nagano. Echo Valley and Tangram Ski Resort.
U2: Can you tell me the price?
S2: Echo Valley is $30 for a one-day pass. Tangram is $27 for a one-day pass.
U3: How far is Echo Valley?
S3: It is 190 kilometers away.
U4: Can you tell me the weather?
S4: **You will arrive there around 1:30 p.m. The weather forecast for that time is sunny**.
U5: Can you tell me the price again?
S5: Echo Valley is $30 for a one-day pass.

Figure 1. Example of dialog context. (U: user, S: system)[1]

To support network and user context, the Reasoner decides what to suggest and when to suggest it. However, suggesting new information to the user may interrupt an ongoing user activity or dialog, and this must be carefully designed to minimize user distraction. This raises three interesting design questions: *what* to say to the user, *when* to say it, and *how* to say it. The i-Finder module in the Reasoner uses contextual events and domain-specific rules to search for information that is appropriate to the user's context (e.g. nearby restaurants, sightseeing attractions and rest areas). The i-Predictor module decides the proper time to provide that information to the user (Ko et al., 2006b).

User adaptation is another important aspect of the context-aware system. CAMMIA maintains a user profile and allows the user to customize system services (e.g. features that are always checked, such as parking availability). CAMMIA also remembers the user's reactions to context-aware suggestions and updates the user profile accordingly. More intelligent user models and adaptation may be addressed in future work.

3. Experimental Setup

The experiment was designed to evaluate our hypothesis that context-awareness has a positive impact on driving behavior when using a dialog system.

3.1. Procedure

Sixteen Japanese subjects (four females and twelve males), ranging in age from 19 to 38 participated in the experiments. The participants were asked to drive at a simulated speed of 100 km/h on a highway course (Figure 2) while searching for restaurants, weather reports and sightseeing information, using both the context-aware and the context-unaware configurations. A baseline test was conducted with eleven out of the sixteen subjects for five minutes.

3.2. Task Description

Task 1: **User context**. The participants had to find a nearby Italian/Japanese restaurant with a parking lot and select it as a trip destination. Some of the restaurants did not provide parking and some would not be open at arrival time. When the participant selected a restaurant that did not satisfy the search conditions, the system provided a notification and returned to the original search result state. The participant then continued to search for another restaurant.

When using the context-aware system, the system notified the participant if the restaurant chosen did not have a parking lot or would be closed, based on simple reasoning using restaurant business hours and the estimated time to the restaurant. When using the context-unaware system, the participants had to figure out parking lot availability and open status by themselves.

Task 2: **Network context**. The participants' task was to find out the weather in Tokyo for that day and the

[1] Nagano is a prefecture in Japan. This dialog example is from Ko et al. (2006a)

following day. To simulate network loss, we used a network loss simulator which made the network unavailable for 20 seconds. When using the context-unaware system, the participants had to monitor the network status and restart the dialog when the network became available. In the context-aware system, the system maintained pending user requests during network loss, and continued the dialog when the network became available and the context was valid (i.e. the participant was not talking to the system and there were no other active dialogs).

Task 3: **Dialog context**. The participants' task was to find a sightseeing site in Shizuoka, a prefecture in Japan for picking oranges, apples, and strawberries. The participants found that the sites would be closed in case of rain and had to ask for weather information in the middle of the sightseeing dialog. Since the context-unaware system overwrote the current sightseeing dialog with a new dialog about weather, the participants had to restart the sightseeing dialog after asking for weather information. The users then checked for other information (e.g. price and business hours) and set the chosen site as a destination in the navigation system. In the context-aware system, the system maintained multiple ongoing dialogs and automatically came back to the sightseeing dialog after the user checked the weather.

4. Results and Analysis

The test results were analyzed using various objective assessments, such as task completion time, the number of user turns and variation in driving behavior. For statistical significant tests, ANOVA (ANalysis Of VAriance between groups) was used. ANOVA tests the statistical significance of the differences among two or more groups whose size is different.

4.1. Task completion time & user turns

All the subjects successfully finished their tasks. Table 1 shows the average completion time and user turns. It took less time and less turns using the context-aware system versus the context-unaware system to complete the required tasks ($p<0.003$).

Context type	Time-to-complete		Turns-to-complete	
	Context-unaware	Context-aware	Context-unaware	Context-aware
User context	217.0 (±76.3)	130.2 (±34.3)	20.6 (±7.4)	12.4 (±2.6)
Network context	46.4 (±8.0)	37.5 (±6.0)	3.3 (±1.7)	1.6 (±1.0)
Dialog context	94.2 (±23.7)	72.1 (±10.6)	10.2 (±2.9)	6.25 (±1.5)

Table 1. Average task completion time (unit: second) & number of user turns

4.2. Analysis of speed

A recent Wizard-of-Oz study conducted in a driving simulator with a speech interface showed that the participants tended to drive more slowly while talking to the system, and the there was no significant changes in lane-keeping which measures the driver's ability to stay in one lane (Geutner et al 2002). To assess variation in driving behavior, we first measured the average speed of the car. Table 2 shows the average driving speed in the three different configurations. The average speed was not significantly different among three groups of test subjects. This indicates that the participants were able to maintain an average speed while using either dialog configuration.

On the other hand, the speed variance differed by the participants and by the configurations, and we measured the statistical significance of the difference. The result shows that the baseline and the context-unaware was different ($p<0.02$) but there was no significant difference between the baseline and the context-aware ($p=0.27$). This indicates that the participants had less speed changes when using the context-aware system.

	Baseline	Context-unaware	Context-aware
Average car speed	25.34 (±0.691)	27.12 (±3.218)	26.84 (±2.288)
Average variance in car speed	0.46 (±0.568)	3.08 (±3.290)	1.79 (±1.242)

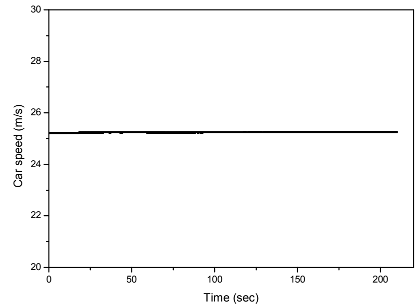
Table 2. Average task completion time

In addition, we measured how frequently a subject changes the car speed. For example, Figure 3 shows the speed changes for one subject in three configurations. The participant had more frequent speed changes in the context-unaware. To measure the frequency of speech changes, we measured total speed changes by adding the speed difference every 0.1 second, and compared the results. This shows that the participants changed speeds more when using the context-unaware versus the context-aware ($p<0.03$).

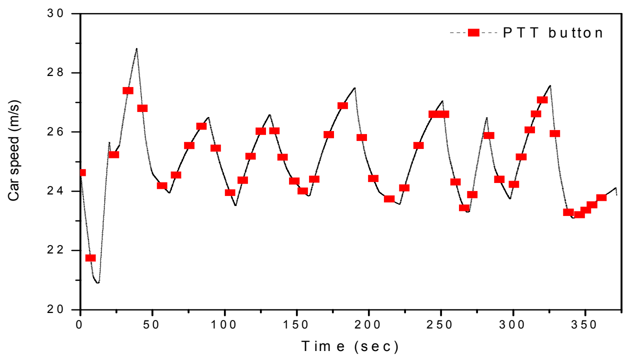
4.3. Analyzing the vector of travel

Another way to measure driving behavior is to compare the angle between the car's direction and the centerline on the road. As the participants drove on a simulated highway under unexceptional conditions, there was no significant difference in the average and the variance. We also measured overall direction change by adding the angle changes every 0.1 second. The results show that the participants tended to change direction more when using the context-unaware than when using the context-aware ($p<0.05$).

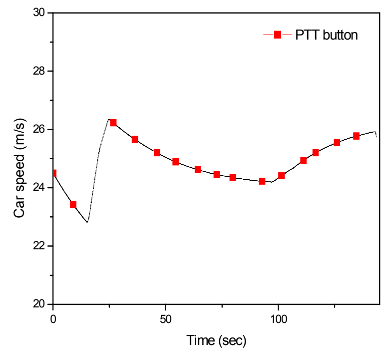
In a different analysis, we measured the changes in angles only on the curved portions of the course. We first identified the parts of the data which were measured along curved parts of the roadway, and then calculated the average of all changes every 0.1 second. The comparison of the averages shows that there was a significant difference between the baseline and the context-unaware ($p=0.0011$) and between the baseline and the context-aware ($p=0.0273$). This indicates that the subjects tended to maintain an optimal driving angle while using either dialog system, but on curved roads, they had more changes of angle when compared to the baseline showing some distraction by the system. Nevertheless, the addition of context-aware dialog features helps to reduce the variance to some degree.



(a) Baseline (25.24 m/s ±0.0001)



(b) Context-unaware(24.99 m/s ±1.86)



(c) Context-aware (24.87m/s ±0.6)

Figure 3. Car speed of one subject in three different configurations.

5. Conclusion

In this paper, we presented the results of an experiment to measure the impact of user-system interaction on the user's driving performance. Our analysis of vehicle speed and the angle of travel showed that context-awareness can reduce the degree of user distraction while driving. The experiment was conducted using a simulated highway course; it is possible that the results may differ when a more challenging course (e.g., local streets with traffic) is utilized, since the user's tolerance for distraction may be reduced in more demanding driving conditions. We are planning to conduct another experiment in a more complex driving environment and using other types of dynamic context change. Experiments which measure user impact during longer-term usage are also left to future work.

6. Acknowledgements

The authors would like to thank Yasunari Obuchi and Nobuo Hataoka for their contributions to the experimental design and evaluation.

7. References

Allen, J. (1995). *Natural Language Understanding*. 2nd ed. Benjamin/Cummings. pp 504-510.

Dey, A. K. and Abowd, G. D. (2000). Towards a Better Understanding of Context and Context-Awareness, In *Proceedings of Workshop on The What, Who, Where, When, and How of Context-Awareness, at Conference on Human Factors in Computing Systems*.

Geutner, P., Steffens, F., and Manstetten, D. (2002). Design of the VICO Spoken Dialogue System: Evaluation of User Expectations by Wizard-of-Oz Experiments. In *Proceedings of LREC*.

Nyberg, E., Mitamura, T., Placeway, P., Duggan, M., and Hataoka, N. (2002). Dynamic Dialog Management with VoiceXML, In *Proceedings of the Human Language Technology Conference*.

Ko, J., Murase, F., Mitamura, T., Nyberg, E., Hataoka, N., Sagawa, H., Obuchi, Y., Tateishi, M., and Akahori, I. (2006a). Robust Multimodal Dialog Management for Mobile Environment. Chapter 22 in *Digital Signal Processing for In-Vehicle and Mobile Systems 2*, Abut, Hansen, and Takeda (Eds.), Springer Science, New York, NY, Scheduled for Spring 2006.

Ko, J., Murase, F., Mitamura, T., Nyberg, E., Tateishi, M., and Akahori, I. (2006b). What, When and How to Communicate: Strategies for Context-Aware Dialogs. *Unpublished manuscript*.

McTear, M. (2004). New Directions in Spoken Dialogue Technology for Pervasive Interfaces. In *Proceedings of Robust and Adaptive Information Processing for Mobile Speech Interfaces Workshop, COLING*.

Obuchi, Y., Nyberg, E., Mitamura, T., Duggan, M., Judy, S., and Hataoka, N. (2004). A Robust Dialog Management Architecture for Car Telematics Systems Using VoiceXML. *DSP in Mobile and Vehicular Systems*. Abut, Takeda, Hansen (Eds.), Kluwer Academic publishers.

Schilit, B., Adams, N., and Want, R. (1994). Context-Aware Computing Applications. In *Proceedings of Workshop on Mobile Computing Systems and Applications*.