Towards Understanding Multi-Tasking Dialogue For Automotive Applications

Fan Yang, Alexander Shyrokov , Peter A. Heeman, Andrew L. Kun

INTRODUCTION

The development of easy to use graphical user interfaces (GUI) has done more to promote widespread use of computers than any other single development. However, GUIs are not appropriate for every situation, especially as we move beyond the desktop. GUIs present information visually and receive instructions from the user in the form of a physical action (e.g. mouse clicks). This makes GUIs inappropriate for situations where the user is performing other eyes-busy and hands-busy tasks, such as driving a car. For automotive applications, speech is a natural interaction mode.

Existing speech interfaces have mostly been used to perform a single task, where the user finishes with one task before moving on to the next. However, we envision that speech interfaces in automotive applications need to work with the user on multiple tasks at the same time, which is especially useful for real-time tasks. For instance, a driver in a car might use a speech interface to catch up on emails, while occasionally checking upcoming traffic conditions, and receiving navigation instructions; or a police officer in a cruiser on duty might need to be alerted to a nearby accident while accessing a database for license plates.

Several speech interfaces that allow multi-tasking dialogues have been built [2, 3]. Kun et al. [2] developed a system called Project54, which allows a user to interact with multiple devices in a police cruiser using speech. The architecture of Project54 allows for handling multiple tasks overlapped in time. For example, when pulling over a vehicle, an officer can first issue a spoken command to turn on the lights and siren, then issue spoken commands to initiate a data query, go back to interacting with the lights and siren (perhaps to change the pattern after the vehicle has been pulled over), and finally receive the spoken results of the data query. While the current implementation of Project54 assumes that the officer initiates the task switching (e.g. the one about lights and the one about data query), the system can initiate task switching too. However, Project54 does not provide infrastructure for signaling to the officer a system-initiated switch. Thus the officer might be confused about which task the interface is talking about and the progress of the task. Multi-tasking dialogues, even in the best circumstances, will be difficult for users, as users need to be aware of task switching and remember the details of each task.

To develop a speech interface to better support automotive applications, we started by understanding how people manage multi-tasking dialogue. Multi-tasking dialogues, where multiple independent topics overlap with each other in time, regularly arise in human-human conversation: for example, a driver and a navigator in a car might be talking about their summer plans, while occasionally interjecting road directions or conversation about what music to listen to.

THE MTD CORPUS AND ANALYSIS

To start understanding multi-tasking dialogue, we created an experimental setup that just involves two verbal tasks. This allowed us to start studying multi-tasking behavior in itself, without the added complications of driving. We collected the MTD corpus [1], which consists of a set of human-human dialogues where pairs of conversants have two types of overlapping verbal tasks to perform: an ongoing task type that takes a long time to finish, and a real-time task type that can be done in a couple of turns but has a time constraint. In the ongoing task, conversants work together to form a poker hand. Each participant has 3 cards, and they takes turns drawing and discarding a fourth card. Conversants cannot see the other conversant, nor the cards in the other conversant's hand. The conversants have to converse to share what cards are in their hands, and what poker hand to try for. Occasionally, one of the conversants has to solve a real-time task by asking whether the other conversant has a certain picture displayed on the screen, which has an urgency reflected in how much time the conversant is given (either 10s, 25s, or 40s).

In our statistical analysis of the MTD corpus, we distinguished three types of discourse context where a player suspends the poker playing and switches to a waiting picture game: (G) immediately after completing a poker game (at the end of a game), (C) immediately after discarding a card (at the end of a card), and (E) embedded inside a card discussion, where players are deciding which card to discard. We found that: (1) conversants try to switch to the real-time task in such a way as to make the switch less disruptive to the ongoing task [1, 6]; (2) conversants mark the switches using discourse markers and prosody, especially with pitch: the more disruptive the switch is, the higher is the pitch [6]; (3) when returning to the ongoing task, conversants sometimes restore the context, by either repeating the previous utterance before the interruption, or summarizing the critical information that was exchanged [5].

IN-CAR MULTI-TASKING DIALOGUE

We have since moved towards understanding in-car multi-tasking dialogue, by using a simulated vehicle [4]. In our experiment, one conversant (the driver) operates a simulated vehicle, and a second (the dispatcher) helps the driver navigate city streets. The dispatcher and the driver cannot see each other and they communicate via headsets. Some city streets are blocked by construction barrels. Thus, the driver could not follow some of the navigation instructions given by the dispatcher, and a dialogue develops as the driver and the dispatcher collaborate to find alternate routes. Periodically the driver is prompted to initiate a short real-time task, which involves informing the dispatcher of a certain visual stimulus. As in the poker-playing task, the prompt includes information about the urgency of the real-time task.

In the statistical analysis, we looked at whether conversants interrupt in the middle of an utterance, in the middle of an adjacency pair, or between adjacency pairs [4]. We concentrated on cases when the driver is prompted to introduce the real-time task during the first part of an adjacency pair, as this is the point in the local discourse structure that is most embedded. We found that conversants either immediately interrupts the first part, or waits until the conclusion of the adjacency pair. This might indicate that they are trying to avoid having the first part of an adjacency pair pending during a task switch, so that there is a simpler discourse context to resume to.

CONCLUSION

This extended abstract gives a brief summary of our ongoing research on multi-tasking dialogue for automotive applications. In the full paper, we will elaborate more details, including the knowledge and lessons that we learned, and some guidelines that we propose for building in-car speech interfaces.

REFERENCES

1. Peter A. Heeman, Fan Yang, Andrew L. Kun, and Alexander Shyrokov. Conventions in human-human multithreaded dialogues: A preliminary study. In *Proceedings of IUI*, pages 293–295, San Diego CA, 2005.
2. Andrew L. Kun, W. Thomas Miller, and William H. Lenharth. Computers in police cruisers. *IEEE Pervasive Computing*, 3(4):34–41, October-December 2004.
3. Oliver Lemon, Alexander Gruenstein, Alexis Battle, and Stanley Peters. Multi-tasking and collaborative activities in dialogue systems. In *Proceedings of 3rd SIGdial*, Philadelphia PA, 2002.
4. Alexander Shyrokov, Andrew Kun, and Peter Heeman. Experiments modeling of human-human multi-threaded dialogues in the presence of a manual-visual task. In *Proceedings of 8th SIGdial*, pages 190–193, 2007.
5. Fan Yang and Peter A. Heeman. Context restoration in multi-tasking dialogue. In *To appear in Proceedings of IUI*, pages ??–??, Sanibel FL, 2009.
6. Fan Yang, Peter A. Heeman, and Andrew Kun. Switching to real-time tasks in multi-tasking dialogue. In *Proceedings of CoLing*, pages 1025–1032, Manchester, UK, August 2008.