Recipes for building voice search UIs for automotive

Martin Labsky, Ladislav Kunc, Tomas Macek, Jan Kleindienst, Jan Vystrcil
IBM Prague Research and Development Lab
V Parku 2294/4, 148 00 Prague 4
Czech Republic
{martin.labsky, ladislav_kunc1, tomas_macek,
jankle, jan_vystrcil}@cz.ibm.com

Abstract

In this paper we describe a set of techniques we found suitable for building multi-modal search applications for automotive environments. As these applications often search across different topical domains, such as maps, weather or Wikipedia, we discuss the problem of switching focus between different domains. Also, we propose techniques useful for minimizing the response time of the search system in mobile environment. We evaluate some of the proposed techniques by means of usability tests with 10 novice test subjects who drove a simulated lane change test on a driving simulator. We report results describing the induced driving distraction and user acceptance.

1 Introduction

The task of designing mobile search user interfaces (UIs) that combine multiple application domains (such as navigation, POI and web search) is significantly harder than just placing all single domain solutions adjacent to one another. We propose and evaluate a set of UI techniques useful for implementing such systems. The techniques are exemplified using a prototype multimodal search assistant tailored for in-car use. The prototype supports several application domains including navigation and POI search, Wikipedia, weather forecasts and car owner's manual. Finally, we report usability evaluation results using this prototype.

2 Related Work

Two examples of multi-modal search UIs for automotive are the Toyota Entune[1] and the Honda Link[2]. Both infotainment systems integrate a set of dedicated mobile applications including a browser, navigation, music services, stocks, weather or traffic information. Both use a tablet or a smartphone to run the mobile applications which brings the advantage of faster upgrades of the in-car infotainment suite. Home screens of these systems consist of a matrix of square tiles that correspond to individual applications.

The answers presented to the user should only contain highly relevant information, e.g. presenting only points of interest that are near the current location. This is called conversational maxim of relevance (Paul, 1975). Many other lessons learned by evaluating in-car infotainment systems are discussed in (Green, 2013).

In recent years, personal assistant systems like Siri (Aron, 2011), Google Now! (Google, 2013) and the Dragon Mobile Assistant (Nuance, 2013) started to penetrate the automotive environment. Most of these applications are being enhanced with driving modes to enable safer usage while driving. Dragon Mobile Assistant can detect whether the user is in a moving car and automatically switches to "Driver Mode" that relies on speech recognition and text-to-speech feedback. Siri recently added spoken presentation of incoming text messages and voice mail, and it also allows to dictate responses. Besides the speech-activated assistant functionality, Google Now! tries to exploit various context variables (e.g. location history, user's calendar, search history). Context is used for pro-active reminders that pop-up in the right time and place. Speech recognition of Google Now! has an interesting feature that tries to act upon incomplete/interim recognition results; sometimes the first answer is however not the right one which is later detected and the answer is replaced when results are refined.

[1] http://www.toyota.com/entune/

[2] http://owners.honda.com/hondalink/nextgeneration

Proceedings of the of the EACL 2014 Workshop on Dialogue in Motion (DM), pages 28–32,
Gothenburg, Sweden, April 26-30 2014.

3 UI techniques to support search while driving

Below we present selected techniques we found useful while designing and testing prototype search UIs for automotive.

3.1 Nearly stateless VUI

While driving and interacting with an application UI, it often happens that the driver must interrupt interaction with the system due to a sudden increase of cognitive load associated with the primary task of driving. The interaction is either postponed or even abandoned. The UI activity may later be resumed but often the driver will not remember the context where s/he left off. In heavily state-based systems such as those based on hierarchical menus, reconstruction of application context in the driver's mind may be costly and associated with multiple glances at the display.

In order to minimize the need for memorizing or reconstructing the application context, we advocate UIs that are as stateless as possible from the user's point of view. In the context of spoken input, this means the UI should be able to process all voice input regardless of its state.

This is important so that the driver does not need to recall the application state before s/he utters a request. For instance, being able to ask "Where can we get a pizza" only after changing screen to "POI search" can be problematic as the driver (1) needs to change screens, (2) needs to remember what the current screen is, and (3) may need to look at the display to check the screen state. All of these issues may increase driver distraction (its haptic, visual and mental components).

3.2 Self-sufficient auditory channel

According to the subjective results of usability tests described in Section 6 and according to earlier work on automotive dictation (Macek et al., 2013), many drivers were observed to rely primarily on the audio-out channel to convey information from the UI while driving and they also preferred it to looking at a display. A similar observation was made also for test drivers who listened to and navigated news articles and short stories (Kunc et al., 2014).

Two recommendations could be abstracted from the above user tests. First, the UI should produce verbose audio output that fully describes what happens with the system (in cases when the driver controls the UI while driving). This includes spoken output as well as earcons indicating important micro-states of the system such as "listening" or "processing". Second, the UI should enable the user to easily replay what has been said by the system, e.g. by pressing a button, to offset the serial character of spoken output. These steps should make it possible for selected applications to run in a display-less mode while driving or at least minimize the number of gazes at the display.

3.3 Distinguish domain transition types

By observing users accessing functions of multiple applications through a common UI, we observed several characteristic transition types.

Hierarchical. The user navigates a menu tree, often guided by GUI hints.

Within domain. Users often perform multiple interactions within one application, such as performing several Wikipedia queries, refining them and browsing the retrieved results.

Application switching. Aware of the namings of the applications supported by the system, users often switch explicitly to a chosen domain before uttering a domain-specific command.

Direct task invocation. Especially in case of UIs having a unifying persona like Siri (Aron, 2011), users do not view the system as a set of applications and instead directly request app-specific functions, regardless of their past interaction.

Subdialog. The user requests functionality out of the current application domain. The corresponding application is invoked to handle the request and then the focus returns automatically to the original domain. Examples include taking a note or checking the weather forecast while in the middle of another task.

Undo. A combined "undo" or "go back" feature accessible globally at a key press proved useful during our usability testing to negate any unwanted actions accidentally triggered.

Figure 1 shows samples for the above transition types using an example multi-domain search assistant further described in Section 4. Similar lists of transition types ware described previously, e.g. (Milward et al., 2006). Based on observing human interactions with our prototype system, we built a simple probabilistic model to control the likelihood of the system taking each of the above transition types, and used it to rescore the results of the ASR and NLU systems.

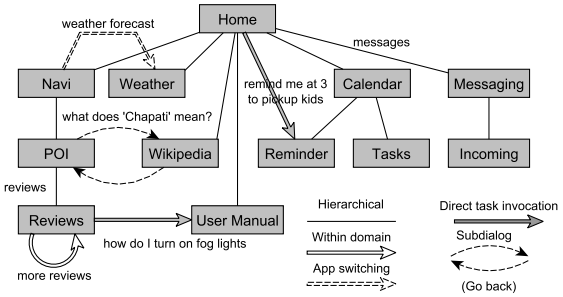

Figure 1: Transitions in a multi-domain system.

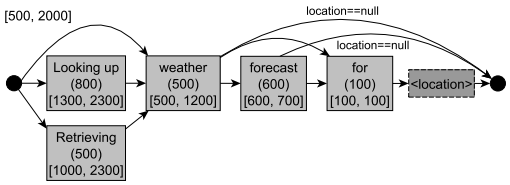

Figure 2: Sample incremental prompt graph. Segments are annotated with durations in round brackets and min/max times before an unknown slot value has to be spoken (ms).

3.4 Early and incremental feedback about the application state

Mobile search UIs often depend both on local and remote resources such as ASR and NLU services and various data providers. In mobile environments, availability and response times of remote services may vary significantly. Most mobile UIs address this problem by responding with a beep and displaying a "processing" sign until the final answer is rendered. We describe a UI technique that combines redundant local and remote resources (ASR and NLU) to quickly come up with a partial meaningful response that addresses the user's request. Chances are that the first response based on partial understanding is wrong and the following prompt must correct it.

Figure 2 shows a template definition for a system prompt that starts playing once the system is confident enough about the user's intent being a weather forecast question. The system provides forecasts for the current location by default but can switch to other locations if specified by the user. Supposing the system is equipped with real-time ASR and NLU that quickly determine the high-level intent of the user, such as "weather forecast", the initial part of the prompt can start playing almost immediately after the user has stopped speaking. While a prefix of this prompt is playing, more advanced ASR and NLU models deliver a finer-grained and more precise interpretation of the input, including any slot-value pairs like "location=London". Once this final interpretation is known, the playback can be directed via the shortest path to the identified variable prompt segments like `<location>`. Further, the selection of prompt prefix to be played can be guided by a current estimate of service delays to minimize chances of potential pauses before speaking prompt segments whose values are not yet known.

4 Voice search assistant prototype

In this section we briefly present a voice search interface that was developed by incrementaly implementing the four UI techniques presented above. While interim versions of this system were only evaluated subjectively, formal evaluation results are presented for the final version in Section 6.

The voice search assistant covers six application domains shown in Figure 3. Navigation services include spoken route guidance together with unified destination entry by voice (addresses and POIs). Some POIs are accompanied by user reviews that can be read out as part of POI details.

Figure 3: Prototype home screen (apps as tiles).

Further, the user can search various knowledge sources like Wikipedia, Wolfram Alpha and the web. The retrieved results are pre-processed and the first one is played back to the user with the possibility of navigating the result list.

To simulate asynchronous events, the system reads out Skype text messages. The driver can also create location and time based reminders that pop up during the journey.

Finally, the system supports full-text search over the car owner's manual. Relevant text passages are read out and displayed based on a problem description or question uttered by the driver.

5 Usability testing setup and procedure

A low-fidelity driving simulator setup similar to the one described in (Curin et al., 2011) was used to conduct lane change tests using (Mattes, 2003). Tests were conducted with 10 novice subjects and took approximately 1 hour and 20 minutes per participant. At the beginning and at the end of the test, subjects filled in pre-test and post-test questionnaires. Before the actual test, each participant practised both driving and using the prototype for up to 20 minutes. The evaluated test consisted of four tasks: an initial undistracted drive (used to adapt a custom LCT ideal path for each participant), two distracted driving trips in counter-balanced order, and a final undistracted drive (used for evaluation). Each of the four drives was performed at constant speed of 60km/h and took about 3.5 minutes. During the distracted driving tasks, the users were instructed verbally to perform several search tasks using the prototype. During task 1, subjects had to set destination to "office", then find a pharmacy along the route, check the weather forecast and take a note about the forecast conditions. Task 2 only differed slightly by having a different destination and POI, and by the user searching Wikipedia instead of asking about weather.

6 Usability testing results

Objective distraction was measured using mean deviation ($MDev$) and standard deviation ($SDLP$) of the vehicle's lateral position (Mattes, 2003). Two versions of both statistics were obtained: overall (computed over the whole trip) and using lane-keeping segments only. The graph in Figure 4 shows averaged results for the final undistracted drive and for the first and second distracted driving tasks (reflecting the order of the tasks, not their types). We observe that using the search UI led to significant distraction during lane change segments but not during lane keeping. Also, the distraction results for the first trip show higher variance which we attribute to the users still adapting to the driving simulator and to using the UI. The observed distraction levels are comparable to our earlier results obtained for a text dictation UI when used with a GUI display (Curin et al., 2011).

Several observations came out of the subjective feedback collected using forms. The users reported extensive use of the auditory channel (both

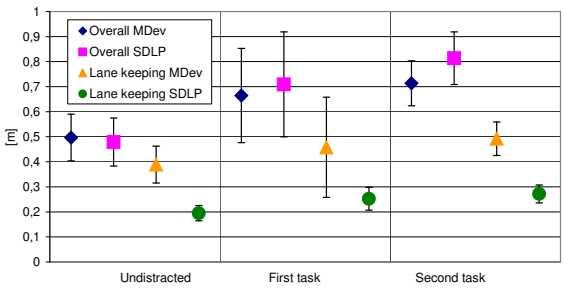

Figure 4: Driving distraction while using a multimodal search UI.

in and out) only with occasional glimpses at the screen (we however observed that objectively they looked at the display more often than they reported subjectively). Users also missed some information in the voice output channel such as audio indication of route calculation progress (which could take several seconds). Reading any text from the screen was found difficult, and users requested that playback be improved; see related follow-up study (Kunc et al., 2014). Interestingly, multiple participants requested voice commands that would duplicate buttons like "next" and "previous", even in cases where speech would be less efficient. This may show a tendency to stick with a single modality as described by (Suhm et al., 2001). Additionally, the users requested better synchronization of navigation announcements like "take exit 4 in 200 metres" with the output of other applications. The baseline behaviour utilized in the test was that high-priority navigation prompts interrupted the output of other applications. Navigation, POI search, simple note-taking and constrained search domains like weather and Wikipedia were found most useful (in this order). Open web search and browsing an original car owner's manual were considered too distracting to use while driving.

7 Conclusion

We described several recipes for building spoken search applications for automotive and exemplified them on a prototype search UI. Early usability testing results for the prototype were presented. Our future work focuses on improving the introduced techniques and exploring alternative UI paradigms (Macek et al., 2013).

Acknowledgement

The presented work is part of an IBM and Nuance joint research project.

References

Jacob Aron. 2011. How innovative is apple's new voice assistant, siri? *New Scientist*, 212(2836):24.

J. Curin, M. Labsky, T. Macek, J. Kleindienst, H. Young, A. Thyme-Gobbel, H. Quast, and L. Koenig. 2011. Dictating and editing short texts while driving: distraction and task completion. In *Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*.

Google. 2013. Google now assistant. Available at `http://www.google.com/landing/now/`.

Paul A Green. 2013. Development and evaluation of automotive speech interfaces: useful information from the human factors and the related literature. *International Journal of Vehicular Technology*, 2013.

L. Kunc, M. Labsky, T. Macek, J. Vystrcil, J. Kleindienst, T. Kasparova, D. Luksch, and Z. Medenica. 2014. Long text reading in a car. In *Proceedings of the 16th International Conference on Human-Computer Interaction Conference (HCII)*.

Tomáš Macek, Tereza Kašparová, Jan Kleindienst, Ladislav Kunc, Martin Labský, and Jan Vystrčil. 2013. Mostly passive information delivery in a car. In *Proceedings of the 5th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '13, pages 250–253, New York, NY, USA. ACM.

Stefan Mattes. 2003. The lane-change-task as a tool for driver distraction evaluation. In *Proceedings of the Annual Spring Conference of the GFA/ISOES*, volume 2003.

David Milward, Gabriel Amores, Nate Blaylock, Staffan Larsson, Peter Ljunglof, Pilar Manchon, and Guillermo Perez. 2006. D2.2: Dynamic multimodal interface reconfiguration. In *Talk and Look: Tools for Ambient Linguistic Knowledge IST-507802 Deliverable D2.2*.

Nuance. 2013. Dragon mobile assistant. Available at `http://www.dragonmobileapps.com`.

Grice H Paul. 1975. Logic and conversation. *Syntax and semantics*, 3:41–58.

Bernhard Suhm, Brad Myers, and Alex Waibel. 2001. Multimodal error correction for speech user interfaces. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 8(1):60–98.