A MOBILE OFFICE AND ENTERTAINMENT SYSTEM BASED ON ANDROID

Felix Burkhardt, Martin Eckert, Julia Niemann, Frank Oberle, Thomas Scheerbarth, Stefan Seide und Jianshen Zhou

DTAG Laboratories
Felix.Burkhardt@telekom.de

Abstract: A system for mobile office and entertainment applications was developed. The input modalities include touch screen and automatic speech recognition, the output modalities screen output and speech synthesis. Services comprise an e-mail reading and responding application, a configurable news feed (RSS) reader and, as entertainment, a license plate information application. Being conceptually eyes- and hands-free, the main target application is the car domain, but as a native android application it runs also on every Android based mobile phone. We report on the software structure and a framework for the integration of speech resources into Android based software, describe the functionality of the services e-mail, RSS and license number information, report on the design concept of multimodal usage, explain integration, evaluation and tuning of speech recognition as well as synthesis and conclude with a description of the evaluation user tests.

1 Introduction

The described system is part of a project, called Connected Live and Drive, which is running in the research and development part of Deutsche Telekom - the Deutsche Telekom Laboratories. This project bundles and combines different knowledge areas and experiences, including HMI (Human Machine Interface) development, software architectures for mobile devices and car specific hardware requirements. The Connected Life and Drive project is focused on the development of multimodal operating concepts for intuitive and largely non-disruptive use of various services during a journey that prevail over the current heterogeneous and proprietary solutions. Various innovative technologies such as voice control, voice output (text-to-speech) and touch screen control have been integrated in order to take account of drivers' needs and of specific car-driving conditions. During service usage drivers hands do not leave the wheel nor their eyes the road, since they operate the entire service by voice control easily and conveniently via controls on the steering wheel. The whole concept is based on the open Android operating system, which software developers and the car industry use to implement and market new applications.

2 Service Description

Using embedded ASR and TTS on the Android platform, both in German and English language, the following services have been implemented as demo show cases. These services can be used via a multimodal user interface.

- **News**: It enables the user to listen to RSS headlines and short news items. The service can be configured by selecting the RSS channels .
- **E-mail**: This use case enables the user to listen, answer, forward, search and delete e-mails. E-mail answering is not yet possible by speech-to-text but by recording voice messages which get sent as e-mail attachments.
- **SMS**: The user can listen, forward, search and delete his SMS via a multimodal user interface similar to e-mail and news service. Answering SMS is not yet possible.

- **License number information**: As an entertainment application, one can speak or type the license letters and retrieve the correspondent county name and plate on the display.

The showcases were realized by Deutsche Telekom in cooperation with Continental to cover car specific applications like seamless navigation as well as the adaptation of the app store idea or entertainment applications for in-car usage.

The applications are meant to enable a safe and enjoyable use of internet and online-services in vehicles. Special emphasis is on closing the communication gap on the road by enabling highly complex communication services like e-mail, SMS or social networking for the user while driving. The realized show cases like news-, e-mail- and SMS-reading demonstrate how the speech recognition and text-to-speech technology contribute to a significant reduction of driver distraction, because the driver isn’t forced to look away from the road any more. In contrast to the “touch only” human machine interface (HMI) the multimodal automotive HMI makes composing e-mail possible while driving. Other comprehensive services like e-mail- and SMS-dictate in car are envisaged based on speech to text technology and might be added at a later stage.

Additionally, the integration of speech recognition helps to simplify applications like music search, address book search and license number search. It makes communication and entertainment much more comfortable and more fun for the driver while driving.

3 Architecture

In order to ease integration of speech technology services like automatic speech recognition or synthesis on Android based devices, a general framework architecture named AVF (Android Voice Framework) was developed. It is based on the Google Android TTS Interface (Google TTS API) description and on the Java Speech API 2 (JSR 113) (Java Speech API).

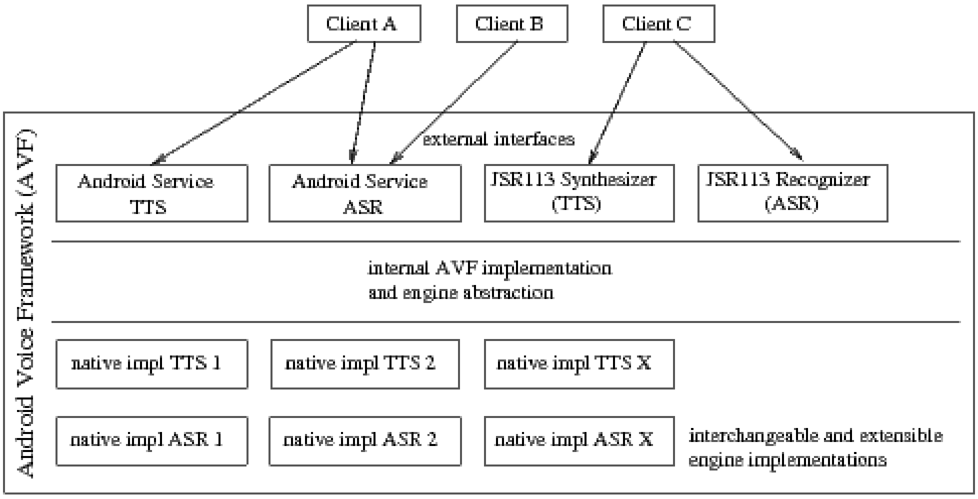

Figure 1: Architecture of speech resource framework

Fundamentally, the architecture is twofold. On the one hand the program interacting with the user, on the other a background service running Text-to-Speech (TTS) and Automatic Speech Recognition (ASR) services. The service also offers an interface to record and play audio files. These services are multi-thread able, i.e. they might be used from several programs at the same time.

The service runs under Android version 1.5, which didn't provide yet for an own TTS service, as well as Android 1.6 with the already integrated TTS service based on Svox TTS version Pico. Since Android 2.2 there is also a simple ASR service already integrated in Android. Our system was ported to all these platforms and can be extended by arbitrary TTS or ASR resources by implementing the interface classes.

In Figure 1, the overall architecture is displayed. Several clients (programs interacting with the user) use several services based on Android service specification or Java Speech API, which are mapped by the AVF to native, proprietary software libraries.

Of course the framework, which is fully Java based, might also be used outside from Android, in case the native libraries were compiled for the target platform.

4 Approaches to User Interface Design

Restricted basically to in-car functionality and entertainment (Radio, MP3, CD) in the past, by now more and more complex functionality like communication, telematics or navigation enters the car. This increasing complexity of in-car functionality causes also increasing requirements to the automotive human machine interface (HMI).

Of fundamental importance for the adaptation of services for in-car usage is the reduction of mental overload and driver distraction in order to avoid crash risk. Ongoing developments in speech recognition, speech-to-text and text-to-speech technology allow adapting the conventional, touch screen based HMI for in-car usage. Two different device classes have been targeted: the integration of a smart phone into the car and a head unit based approach. Figure 2 displays screenshots of both interfaces.

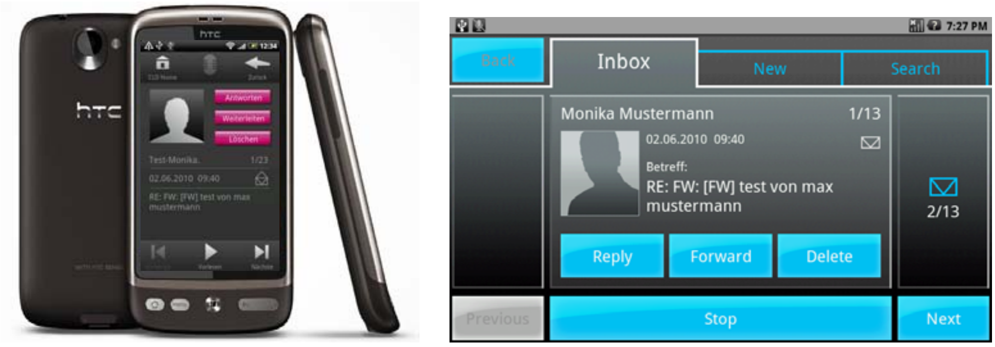

Figure 2: Smart phone and head unit interface of e-mail service

This section will outline how to build a multimodal interface, which is able to adapt services for in-car usage with simple but intuitive user interfaces and an adapted presentation.

4.1 Input and output modality components

The described multimodal approach for an automotive HMI is based on a voice user interface (VUI), focused to the driver, while driving his car, and a graphical user interface (GUI), based on a touch screen and intended to be used in calm traffic situations, in the parked car or by the co-driver. Both input modalities, speech commanding as well as pointing gestures on the touch screen, are for the user at any moment available. A push-to-activate control on the steering wheel offers easy access to speech recognition. Acoustical system feedback ensures that the driver isn't forced to put his eyes away from the road.

4.2 User driven interaction

The user is able to interrupt acoustic system prompts whenever he wants in order to enter his voice command by a push on the push-to-activate control on the steering wheel. But the user isn’t forced to answer a system prompt immediately. If the user has lost orientation about the state of interaction, because he has interrupted the interaction for a longer time, he is able to pick up the current task by executing a long term push on the push-to-activate control on the steering wheel. As reaction the system plays the context sensitive information about the current interaction step, where the interaction previously stopped. Furthermore the context sensitive information will be presented automatically whenever speech recognition fails.

4.3 Design guidelines

Narrow menus, i.e. not more than 5 options per graphic, have to contribute to a reduction of mental overload and driver distraction. Graphical interaction components like buttons or fields have to reflect the associated voice commands. Feedback is given to each input of the user. As reaction to a speech commands feedback and user guidance will be given by acoustic and graphic system output. As reaction to pointing gestures on the touch screen feedback and user guidance will be exclusively given by the touch screen.

The audio prompting gives feedback about the state of the interaction and emphasizes the keywords, which mirror the current view of the touch screen. The audio prompting is as short and crisp as possible, in order to avoid annoyance in daily usage. The user of an in-car HMI is aware of the fact that he is speaking to a machine. Therefore speech recognition is focused on the recognition of single words and very short phrases, thus also contributing to an enhancement of robustness of speech recognition in the noisy in-car environment.

Using the input modality speech, the user is able to select the requested task directly by saying the according voice command without taking the long way round the menu structure. Universal fall backs “Home” and “Back” are available for the case that interaction is deadlocked.

The described multimodal HMI allows the user to interrupt the human machine interaction at any time, when the traffic situation might require the driver’s full attention, and to continue the interaction later, when the traffic situation is relaxed again. Based on a user driven interaction model, the VUI is able to play the leading part for the user, who is occupied with driving his car, and leads to a significant increase of comfort and reduction of mental overload and driver distraction.

5 Speech Recognition

A commercial ASR engine suitable for embedded platforms and especially automotive dashboard applications has been used to perform speech recognition and is integrated into the system by the afore mentioned Android Voice Framework (see chapter 3). The possible integration of a second ASR bidder is in preparation but not yet finished, performance comparison tests will than be carried out.

Acoustic models for the 16 kHz sampling frequency are provided by the vendor together with the SDK. Additionally these models are optimized for in-car environments. Speech recognition for our application was integrated for two languages: English and German

The grammars were written in a BNF like grammar format that is supported by the engine. Although the main scope of the ASR is to recognize short commands (e.g. “e-mail” or “news”) also natural language understanding (e.g. “search for e-mails from Marcus”) is supported. The second, not yet integrated, ASR maker uses grammars in JSGF format. The conversion is done automatically based on an AWK script.

Based on the assumption that the user commands rely on the wording displayed on the graphical user interface (GUI) there is a strong correlation of GUI and VUI (voice user interface).

The starting point for tuning the recognition accuracy was manual transcription. Those words that could not be found in the system dictionary provided by the vendor were added together with their phonetic transcriptions into an application specific user dictionary. A wizard-of-Oz test that might show what prospective users might say has not yet been carried out.

To recognize the letters of the city/region prefix code in the license plate application we added a spelling grammar with 425 entries of one up to three letters (e.g. “d”, “d e” or “d e l”).

An evaluation of recognition performance for German version of the ASR was made based on 9.078 samples. Because data from the application wasn’t yet available, a comparable (with respect to a command and control situation) dataset from a former project was used. In this data, about 30 users control their television with a voice interface. The test samples contain German TV stations (e.g. “MTV Music”, “Sat 1” or “Das Erste”), categories (“science fiction”, “comedy”) and commands (“sound off”, “main menu”).

Because this data was recorded in the participant’s living room, but the target application will mostly be used in a car, they were additionally mixed with street noise (from an ITU codec test suite), using a factor of 0.5. We didn’t use a common dB threshold for each sample because we think samples with a constant amount of noise but varying amount of information are more adequate for a real word situation.

The recognition test was performed on all 9.078 samples using a grammar with about 150 entries (TV stations, categories etc.). We decided to compute Command Success Rate (CSR) instead of Word Error Rate (WER) because for some of the TV stations there is more than one variant (e.g. “Erstes” and “das Erste”), and in contrast to speech-to-text services, the application is not about the exact wording but about recognizing commands.

	Clean Speech	**Car Noise Speech**
# test samples	9.078	9.078
# correct results	4.380	2.484
command success rate	**48,25%**	**27,36%**

Table 1: Evaluation results of ASR tests

For clean speech, a command success rate of 48,25% was computed. Results on the utterances with car noise show a command success rate reduction by more than 20%, i.e. has a strong influence.

This does not mean that these results are expected for the target application because the data is too different. We used consciously a very high SNR in order to see differences between different ASR maker more clearly. As stated, these tests will be repeated with different ASR engines and the results than be used primarily for a comparison.

6 Speech Synthesis

Speech synthesis was also integrated with the afore mentioned Android Voice Framework (see section 3). Speech synthesis technology can be grouped in the following techniques (Dutoit, 2001), sorted in historical order.

- **Formant-synthesis**: speech synthesized by physical models (formants are resonance frequencies in vocal-tract). Very flexible and smallest footprint, but very unnatural due to insufficient models.
- **Diphone-synthesis**: speech concatenated from diphone-units (two-phone combinations), prosody-fitting done by signal-manipulation (depends on unit-coding). relatively small footprint but not very natural.
- **Non-uniform unit-selection**: best fitting chunks of speech from large databases get concatenated, minimizing a double cost-function: best fit to neighbour unit and best fit to target prosody. Sounds most natural (similar to original speaker), but inflexible with respect to out-of-domain words and large footprint.

In this project, non-uniform unit-selection was used because it clearly sounds most natural and it would be hard for customers to accept lower quality. In order to find the best quality from several bidders with respect to our application, we performed several listening tests:

- In order to test the out-of-the-box quality, several hundred words from the news and license plate domain (German county names, actor names, country names) were synthesized by competing text to speech synthesizers without adding in-domain pronunciation lexica, and rated by an expert listener with respect to pronunciation errors.
- To test the overall naturalness of the engines, several ten in-domain texts (private and business e-mails, short news items from politics and culture) were synthesized and rated informally by a group of 5 expert listeners.

Due to time restrictions, both tests were only done for the German version and not repeated for the English version.

Speech synthesis depends highly on comprehensive pronunciation lexica, therefore in-domain tuning is almost always a necessity. This was done based on the RSS feed reading application. An expert listener used the application in a time period shortly before the main presentation (at CeBIT) and noted pronunciation errors which later were manually transcribed and added to the user lexica.

A further expert listener experiment was carried out to test whether users prefer prompts that mix static content by human speakers with dynamic content from speech synthesis or prefer a voice user interface that sticks to one voice by using speech synthesis in the whole voice user interface.

Because the expert listeners preferred the solution based solely on speech synthesis, all system prompts were generated beforehand and the pronunciation manually enhanced if necessary by adding entries to the user lexica.

7 Evaluation

Whereas most of the services for smart phones are developed and designed for a ‘non-specific’ usage scenario, the CLD applications were specially developed for the use within the car – while driving. This poses a large challenge towards the design of the interface. In addition to the general requirements for the service usage, the distraction for the driver while using the service has to be kept as low as possible. Driving as the primary task has to be in focus of driver's attention at almost all times and should not be interrupted by interacting with other systems such as the infotainment system.

As a consequence, an intuitive and safe interface for mobile service used while driving has to be developed. Therefore a user-centred design process was applied accompanied by an iterative testing during the different design stages.

7.1 Evaluation with experts

The first user tests took place during the early stage of the system design. Six usability experts were asked to evaluate the preliminary screens and the interaction flow of an e-mail application by paper prototyping. According to Nielsen (1994) not more than five test subjects are necessary to detect major design problems. In a wizard of oz scenario, one of the designers simulated the system by changing the paper screenshots and calling the specific speech prompts after each input by the test subject. The experts performed different tasks and were asked to say out loud what they are doing and thinking while interacting with the paper prototype. This thinking aloud technique is used to observe problems, actual errors and surprises directly in each interaction step (Lewis, 1982). The test was recorded and has been evaluated afterwards. If there were problems detected by more than half of the experts, the interaction concept was adjusted.

Before the driving simulator experiment has started, another usability test with six subjects was conducted to evaluate the first revised version. Again, the thinking aloud technique had been used based on a click-dummy prototype where the subjects had to fulfil certain tasks. All participants of this pre-test were novices. The testing showed that not only usability experts are able to interact with the system but also 'normal' users.

7.2 Driving simulator study

The main usability test focused on the distraction from driving caused by the system and was conducted as a driving simulator study with 56 test subjects. The test setting of this study was selected in order to simulate the secondary task paradigm of in-car infotainment system use as well as to measure how much interacting with the application influences the performance of the primary task driving (e.g. variance in the lane position). Additionally, eye movements (percent dwell time on the road and percent dwell time on the smart phone screen) with a head mounted eye tracking system and furthermore the input duration and error rate while performing the infotainment task was logged. We also quantified mental workload caused by the system and acceptance of the system by using standardised questionnaires. The mental workload was measured by the NASA TLX (Hart & Staveland, 1988) and the acceptance by the product acceptance questionnaire (Davis, 1989).

Three different versions of the application's voice user interface were compared to each other on these parameters. One of the voice user interfaces was designed following a cognitive model of attention allocation in dynamic environments (SEEV-model, Wickens, 2002) in order to reduce visual distraction while driving. In spite of voice user interfaces of in-car infotainment systems, there is a tendency of drivers to spent less time looking on the road if a graphical user interface is available than without such a display. It was assumed that these effects are caused by a high time effort and a low expectancy of speech output. To reduce the probability of attention allocation towards the displays of in-car infotainment systems, we increased the expectancy and the bandwidth of auditory information for the e-mail application. The application provides the driver the opportunity to request an auditory representation of the information given on the screen at any time. To decrease the time effort we shortened the speech output by using nonverbal sounds (like earcons and auditory icons), up-tempo speech of information that was already heard, and by using different voices to represent meta-information. It was hypothesized that the design recommendations based on the cognitive model of attention allocation would lead to less looking on the display of the smart phone.

Not only the versions of the voice user interface got varied also the influence of different driving task was tested. Therefore the degree of visual demand caused by the traffic situation was affected.

Finally the subjects were asked to perform two different kinds of tasks with the e-mail application. One of the tasked was well trained; one of them was new, whereby the interaction steps were not known. The analysing of the data is still in the process, as we are not able to present exact results yet, but the hypothesis that the measures to reduce driver's distractions help can be confirmed.

8 Summary and Outlook

The developed concept and implemented applications enable a safe and enjoyable use of internet and online-services in vehicles. The key challenge, to bring more applications to the car which are usable during drive and understandable without guidance, is a well designed user interface. Within the Connected Live and Drive project such user interface was developed, focussing the goal to keep the handling simple and intuitive, supported by state of the art voice technologies like speech recognition and high quality text to speech added by touch screen interactions.

The concept idea of Mobile Office and Entertainment System based on Android was implemented on different Android based devices like smart phones and a car specific entertainment unit (head unit). During CeBIT 2010, the whole system, running on a head unit in a VW Passat car, was demonstrated at the Deutsche Telekom booth in cooperation with Continental and T-System to a very large audience.

The current system is under further development to integrate additional services and enhance the concept for a wider range of mobile devices.

Literature

[1] Davis, F.D. (1989). Perceived Usefulness, Perceived Ease of Use and User Acceptance of Information Technology. *MIS Quarterly*, 319-340.

[2] Dutoit, T. An Introduction to Text-to-Speech Synthesis, Springer 2001

[3] Google Android TTS API http://developer.android.com/reference/ android/speech/tts/TextToSpeech.html

[4] Hart, S.G.. & Staveland, L.E. (1988). Development of a multi-dimensional workload rating scale: Results of empirical and theoretical research. In P.A. Hancock & N. Meshkati (Hrsg.), *Human mental workload*, 139-183. Amsterdam (NL): Elsevier.

[5] Java Community Process: Java Speech API JSR 113 http://jcp.org/en/jsr/detail?id=113

[6] JSGF Java Speech Grammar Format http://java.sun.com/products/java-media/speech/forDevelopers/JSGF/

[7] Lewis, C. H. (1982). Using the "Thinking Aloud" Method In *Cognitive Interface Design*. Technical Report IBM RC-9265

[8] SVOX TTS http://www.svox.com/

[9] Wickens, C.D. (2002). Multiple resources and performance prediction. *Theoretical Issues in Ergonomics Science, 3*(2), 159-177.