# Speech Dialog Generation from Graphical UIs of Nomadic Devices and the Integration into an Automotive HMI

Sven Reichel
Ulm University
Institute of Media Informatics
Ulm, Germany
sven.reichel@uni-ulm.de

Ute Ehrlich
Daimler AG
Research and Development
Ulm, Germany
ute.ehrlich@daimler.com

Michael Weber
Ulm University
Institute of Media Informatics
Ulm, Germany
michael.weber@uni-ulm.de

## ABSTRACT

More and more people use smartphones regularly for various tasks. Due to distraction issues, the usage is prohibited while driving and thus an integration into the automotive HMI is needed. As speech interaction distracts less than visual/haptic interaction, the smartphone integration needs to support the speech interaction concept of the automotive infotainment system. This paper presents a method to generate the lexicon, grammar, and dialog flow for the car's Speech Dialog System (SDS) based on the GUI specification of smartphone applications. Our approach is platform-independent and application-independent, provides consistent dialogs for all smartphone apps, and complies with the car's interaction paradigm.

## Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*GUI; Natural language; Voice I/O*

## Keywords

Modality translation, multimodal software development, synonym phrases, task-based interaction

## 1. MOTIVATION

Nowadays, people would like to use nomadic devices even while driving. Therefore, automotive HMIs provide connections to phones, mp3 players, and smartphones. However, only the basic functionality of these devices is available in the HMI and people tend to use smartphones despite it is prohibited and dangerous. Many solutions evolved which integrate the smartphone's UI in the automotive HMI and allow a visual/haptic interaction (e.g. MirrorLink[1]). However, voice control is neglected. On smartphones, there are applications (apps) which support voice control (e.g. Apple's Siri). They could be used and integrated into the automotive HMI. However, not all functions can be controlled by voice - especially, third party apps are neglected.

In general, to develop a multimodal app two approaches are common: specifying each modality or defining the UI with modality-independent models. Manual specification for multiple modalities is time-consuming and model-based development requires special knowledge by the developer. Furthermore, for apps on smartphones the visual/haptic modality works fine but from the developer's point of view it does not pay off to provide multiple modalities. As a result, the speech modality is missing by integrating the smartphone into an automotive HMI. Adding this manually is not possible due to the quantity and open application scope of third-party apps.

So far, no domain-independent integration of nomadic devices into an automotive HMI considering speech modality exists. To resolve this and allow voice control of smartphone apps, this paper shows work-in-progress to extract user tasks from the GUI and to generate speech dialogs based on the tasks. As GUIs of smartphones overlap with websites to some extend, [2, 6, 3] form a basis for our work.

[1] `http://terminalmode.org/`


*AutomotiveUI'12*, October 17-19, Portsmouth, NH, USA.
Adjunct Proceedings

## 2. TRANSLATION FROM VISUAL/HAPTIC MODALITY TO SPEECH MODALITY

In the development process of an app the developer considers the user's tasks as well as the app's functions and creates a UI providing the interaction possibilities. On a smartphone the UI consists of various GUI widgets which are assembled logically in a hierarchical structure. Each of the widgets has specific functions which support the users to complete their tasks. We analyze the elements on the GUI in terms of attributes, affordances, and relationships to each other to derive from the elements a set of tasks the user can perform with the GUI. We use the tasks to translate the visual/haptic modality into speech dialogs. As each modality requires an adaptation of interaction elements to supply an efficient usability [4], we have chosen to first abstract the GUI to task level and second reify the tasks to speech dialogs (see Figure 1). This process complies with the CAMELEON Reference Framework (CRF)[1].

The abstract GUI elements are based on Simon et al.'s classification [5], however, are refined by considering the user tasks. This results in the abstract GUI elements: information, multimedia, state, action, input, and structure. Each platform-dependent GUI widget can be abstracted with these types. In the first step of the translation a platform-dependent GUI description (Android XML) is abstracted to User Interface Markup Language (UIML). The UIML file contains the assembling of the GUI in abstract GUI elements including important attributes like size, color, emphasis, grammar, and ontology. For example, a TimePicker (widget for selecting the time of day) in Android is abstracted to an input element referencing the time ontology. Each abstract ele-


**Figure 1: Translation steps from visual/haptic modality to speech modality**

**Figure 2: Semantic of TextField is defined by Label (Screenshot)**

ment stands for various tasks. For example, an input element allows the insertion of text and presents the inserted text to the user. Furthermore, for each task meta speech dialogs are specified which allow task execution by speech. In summary, the GUI is translated into meta speech dialogs which are instantiated with application data at runtime.

# 3. INSTANTIATION OF SPEECH DIALOGS WITH DYNAMIC DATA FROM APPS

The meta speech dialogs are instantiated with data from the corresponding GUI element on the smartphone. Based on this, vocabulary and grammar for the SDS are generated. The grammar includes in each phrase a user task and its semantic GUI reference. The semantic GUI reference is the element users associate with the task. The abstract GUI element which can fulfill the user's task differs from the one providing the semantic information. Considering the Western Culture with reading direction from left-to-right and top-to-bottom, a GUI element sets the semantic for its following ones. For example, in Figure 2 the Label's description ("Next Meeting") assigns the semantic for the following input elements ("2012/10/17" and "Portsmouth"). Our matching algorithm processes a UIML GUI description and identifies a GUI element which can fulfill the user's task based on the element providing the semantic information. An example phrase for the GUI in Figure 2 is: "Set *Next Meeting* to *2012/10/17*".

So far, only runtime values and meta phrases are considered which result in an enhanced "say-what-you-see" system. In natural language different phrases can have the same meaning (synonym phrases). We address this by assigning ontologies to abstract GUI elements and thus allow activation of special grammars (e.g. time grammars). For dynamic data, which is not known until runtime, we use pattern matching to determine the ontology. Furthermore, a thesaurus looks up synonyms. These methods are combined to generate various synonym phrases for each original phrase and are added to the SDS's lexicon. A synonym phrase for the GUI in Figure 2 is: "Set *Next Conference* to *Wednesday*".

Keeping dialogs short, the output depends on the importance of the GUI element providing the data. The user is primarily interested in the most important fact and thus this is read out. Less important information can be accessed by explicit request. The significance of a GUI element is calculated based on its appearance, namely size, color, and emphasis. This means in Figure 2 the bold *2012/10/17* is more important than *Portsmouth*, which results in the dialog: "What is the content of *Next Meeting*?" - "2012/10/17".

# 4. EVALUATION

For evaluation purpose, we implemented our algorithms in a prototype based on Android and the Daimler's SDS (the SDS provides speech understanding, dialog handling, TTS, and simulation of an automotive head unit). As input elements can require arbitrary text, a hybrid Automatic Speech Recognition with a local grammar-based speech recognizer and a cloud-based dictation is used. Two smartphone apps demonstrate the technical feasibility and application-independence of our method (the video[2] shows a sample dialog with the calendar app). Natural dialogs and usability was neglected and is a matter of ongoing research.

# 5. CONCLUSIONS AND FUTURE WORK

This work shows the technical feasibility of a semi-automatic translation method from a GUI to a voice UI based on the CRF. All functions of the GUI are accessible by speech and are adapted to the characteristics of the speech modality. The abstraction of a GUI to UIML guarantees platform-independence for our translation method. However, for each platform the widget set needs to be transformed into abstract GUI elements. Furthermore, using dynamic data ensures app-independence, but requires a data exchange between GUI and SDS. Due to the meta speech dialogs the voice interaction is consistent for all apps and can be adapted to the interaction paradigm of the automotive HMI. The technical feasibility and simplification of multimodal software development has been proven by the prototypical implementation. The next steps focus on the user, which means task-oriented interaction with natural dialogs and an evaluation with user participation in which the driver distraction, task success, and usability will be tested.

# 6. REFERENCES

[1] G. Calvary et al. A Unifying Reference Framework for Multi-target User Interfaces. *Interacting with Computers*, 2003.

[2] L. Paganelli and F. Paternò. Automatic Reconstruction of the Underlying Interaction Design of Web Applications. In *Software Engineering and Knowledge Engineering*, New York, 2002.

[3] F. Paternò and C. Sisti. Deriving Vocal Interfaces from Logical Descriptions in Multi-Device Authoring Environments. In *Web Engineering*. Berlin, 2010.

[4] S. Carter et al. Dynamically Adapting GUIs to Diverse Input Devices. In *SIGACCESS on Computers and Accessibility - Assets '06*, New York, 2006.

[5] R. Simon, M. Jank, and F. Wegscheider. A Generic UIML Vocabulary for Device- and Modality Independent User Interfaces. In *World Wide Web Conference*, New York, 2004.

[6] Z. Sun et al. Dialog Generation for Voice Browsing. In *Building the mobile web: rediscovering accessibility?*, New York, 2006.

[2] `youtube.com/v/gdHDhhNfvvk`