User Interface Management Cheat Sheet

by [deleted] via cheatography.com/2754/cs/18042/

Introduction

The presentation and capture of user information is being isolated from the computational aspects. Could this be the beginning of user interface management systems (UIMSs)?

Source: http://www.speechtechmag.com/Articles/Column/Forward-Thinking/Is-App-Development-Moving-Toward-User-Interface-Management-Systems-128212.aspx

1. Responsive design

With the rapid adoption of devices with displays of varying sizes, applications must present results on everything from smart watches to large, wall-size screens. The goal of responsive design is to recognize the size and capability of various displays so that information can be presented in formats compatible with each device. This includes information presented on devices equipped with only a speaker, using speech synthesis.

2. Flexible capture software

Mobile device application developers often reuse software that allows users to choose the method for entering data. Users can type, select, click, swipe, touch, or speak to enter information.

3. Question/answer dialogues

By using VoiceXML as an intelligent API, users can speak and listen to a question/answer-style conversation that produces a string of characters, which are returned to the underlying application. VoiceXML can also handle conversations using text messages.

4. Conversational-style user-computer dialogues

Chatbots are trained using machine learning to recognize the intentions of users from spoken words and texts and provide the underlying app with the corresponding commands and parameters. Software agents based on these technologies use short, natural conversations through smart speakers.

The UIMS To-Do List

These UIMSs should be able to do the following:
1. Select the presentation format and style best suited for the user and enable the most convenient technique for entering information based on user preferences, usage history, environmental considerations, and available devices.
2. Incorporate state-of-the-art techniques and designs
3. Use standard APIs for transmitting information among user interface components and the back-end applications.
4. Use micro-services for providing specialized functions, including automatic speech recognition (ASR), text to speech (TTS), dialogue management, vision, orientation, and location, while supporting new micro-services such as emotion detection and lip reading as they become available.
5. Support the gradual evolution of users from novices to experts as they become more proficient with the UIMS.
6. Provide consistency in user interface style across different user interface modes and platforms, so users can apply their knowledge of one user interface to another.

Stumbing Block

A major trend in app development today could prove a stumbling block to UIMSs: using the APIs of micro-services within applications. While this enables a wide variety of new and useful applications, the applications themselves are bound to the micro-services, which makes it difficult to update a micro-service or replace it with one from another vendor. Using micro-services in this way will entangle the user interface with the computational logic of applications. This is counter to the philosophy of having a UIMS isolate the user interface from computational logic..