Technical Disclosure Commons

Defensive Publications Series

April 06, 2017

Feedback System For App Developers To See Most In-Demand Voice Command Actions That Could Be Added To Their App

Tuna Toksoz

Thomas Price

Follow this and additional works at: http://www.tdcommons.org/dpubs_series

Recommended Citation

Toksoz, Tuna and Price, Thomas, "Feedback System For App Developers To See Most In-Demand Voice Command Actions That Could Be Added To Their App", Technical Disclosure Commons, (April 06, 2017)
http://www.tdcommons.org/dpubs_series/461

This work is licensed under a Creative Commons Attribution 4.0 License.
This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

FEEDBACK SYSTEM FOR APP DEVELOPERS TO SEE MOST IN-DEMAND VOICE COMMAND ACTIONS THAT COULD BE ADDED TO THEIR APP

ABSTRACT

Systems and methods that allow users to submit requests for a voice action to be added to an app are provided. The system includes a mobile computing device with a voice command service and one or more installed apps connected to an app store. The method includes allowing the user to submit a request for an unsupported voice command, which is conveyed to the developer of the app through the app store. The developer receives structured user data of the app as an ordered list of the most highly in-demand voice actions to be considered for integration through the voice command APIs. The method allows app developers to improve functionality of voice commands easily and to be aware of what the users are trying to accomplish via voice command in their apps.

BACKGROUND

Users may wish to use voice command to control the functionalities of an installed application (or app) in their mobile computing devices, such as an app developed by a third party. However, such a user requirement remains unmet as the app developers will need to add API interactions to tell the voice command service on the mobile computing device exactly what commands their app should handle. The app developers are not aware of what the users are trying to do via voice command in their app. For instance, a popular music app developer may not be aware of the phrases being spoken to the voice command service in order to activate the play function of the music app. Further, users typically have a variety of apps installed in their mobile computing device, such that it remains difficult for the company developing the voice command service of the mobile computing device to hand-code the API interactions for each app. Furthermore, it is desirable for the companies developing the voice command service of the mobile computing device to be seen as impartial and not favoring some apps over others.

DESCRIPTION

Systems and methods that allow users to submit requests for voice actions to be added to an app using a voice command service on a mobile computing device. The system is illustrated in FIG. 1 and includes a voice command service and one or more apps installed in a mobile computing device, interacting with an app store that provides feedback to app developers. The voice command service may be an intelligent personal assistant capable of receiving a voice action from the user and storing it as user data. The mobile computing device may be based on any mobile operating system. The mobile computing device is connected to an app store which relays the user data to an app developer. The system may also include a user interface (UI) embedded in the normal set of UI tools for conveying feedback to the developer.

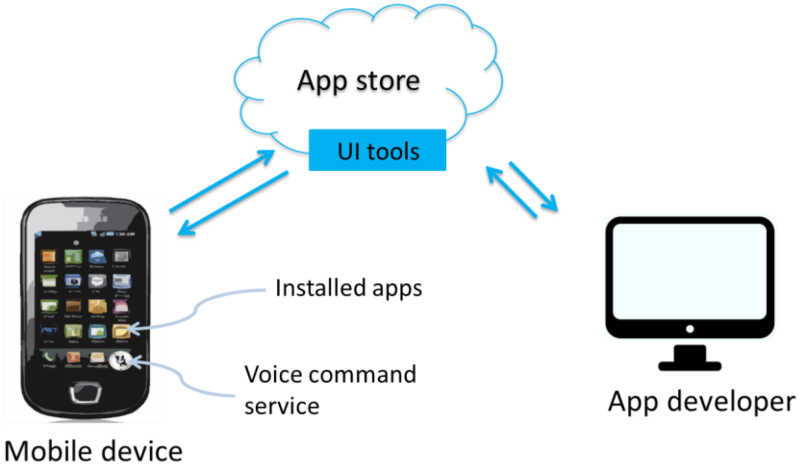

FIG. 1: A system of providing voice command action feedback for app developers

The method as illustrated in FIG. 2 includes recognizing a voice command received from a user and classifying it as either supported or unsupported by the current app. Upon receiving an unsupported voice action, the user is prompted to send a request to the developer of the app to enable support for the requested voice action. The user is sent a warning about the potential use of the voice data by third party developers. The method includes saving the voice action, storing the user request for the selected app in the device and sending it to the app store. The voice action may be further parsed into the core words plus specific detail data before sending the feedback to the app store. The app developer receives the user data through UI on the app store. The app developer is presented with a list of the most highly in-demand voice actions to be considered for integration with the available voice command APIs. Alternatively, the method allows users to request fixes to the voice integrations of their apps. Upon receiving a supported voice action, the action is executed by the voice command service. The user is then prompted for a feedback on the execution to determine if the user is satisfied with the app behavior. If the users say "no" then the steps as illustrated in FIG. 2 will follow as explained above, so that the user may request that the correct voice command feature be added by the app developer.

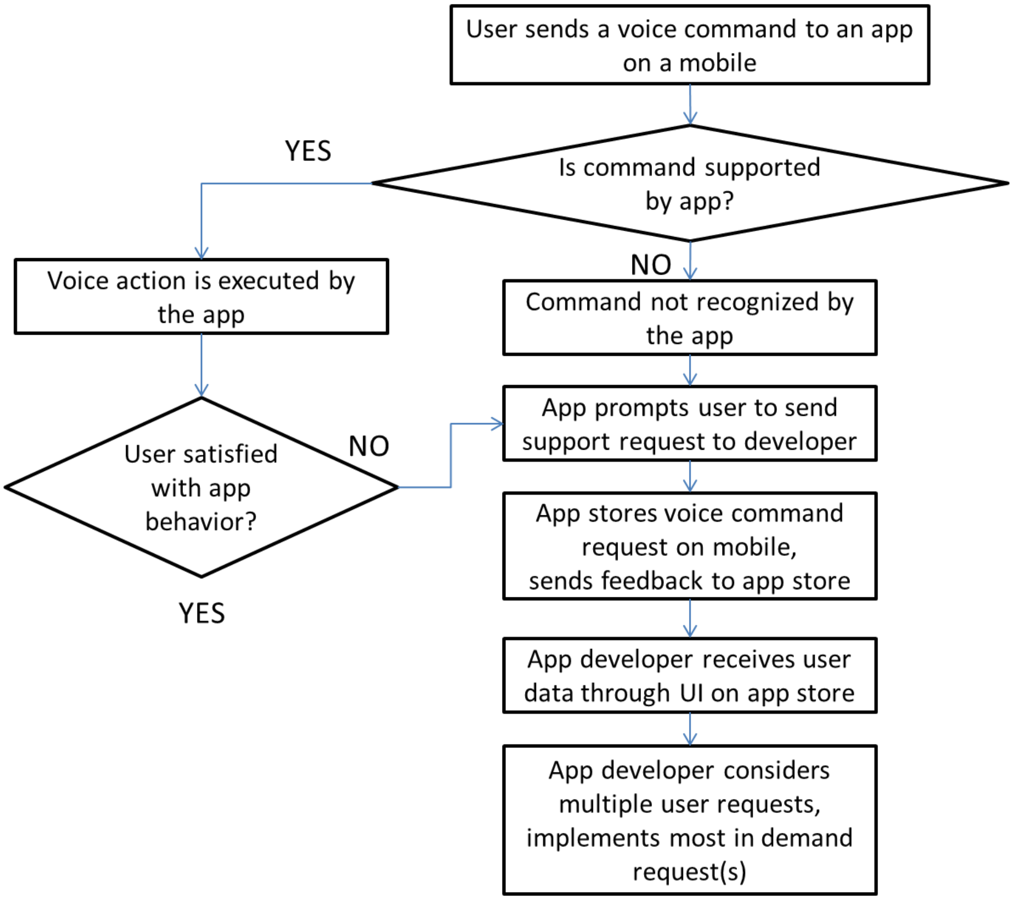
User sends a voice command to an app on a mobile
Is command supported by app?
YES
NO
Voice action is executed by the app
Command not recognized by the app
User satisfied with app behavior?
NO
YES
App prompts user to send support request to developer
App stores voice command request on mobile, sends feedback to app store
App developer receives user data through UI on app store
App developer considers multiple user requests, implements most in demand request(s)

FIG. 2: A method of providing voice action feedback for app developers

An example for the system and method is as follows: A user issues an unsupported voice command to the voice assistant X on their mobile device with a multimedia streaming application Y, “Hey X, play me my summer playlist on Y". Instead of a generic response such as "Sorry I do not understand", the voice assistant prompts the user to request this action from the developer of Y: "The apps on your phone do not support that voice command yet - would you like to submit a request for it?" If the user says "yes" he is asked to select which app on his device should handle this voice command. For privacy, the user may be warned that the voice data from this command might be shared with a third party app developer. Using data from the platform's app store, the device is then able to save this command and store it as a request for the app in question.

In another example, a user issues a voice command to the voice assistant X on their mobile device with a multimedia streaming application Z by saying "Hey X, play me my summer playlist on Z". The Z media app may attempt to handle the command, but it may not support playlists, or it may not understand the semantics of "my summer playlist". The voice assistant system prompts the user, after executing their voice command "did this do what you wanted?” When the user answer "no" then the logic mentioned above is triggered, so that the user may request that the correct voice command feature be added.

The systems and methods allow for presentation of structured user data to the developer of the app - through a UI. The UI may present an ordered list of the most highly in-demand voice actions. This allows the app developer to consider the most highly in-demand voice actions to be added via integration with the available voice command APIs. The systems and methods make the app developers aware of what the users are trying to do via voice command in their app. Furthermore, it allows for the company developing the voice command service of the mobile computing device to notify app developers for improvement in functionality of voice commands easily and without favoring some apps over others.