Platform-Agnostic Voice-Operated Home Automation App for Enhanced-Living of Elders and Those with Disabilities

Monica Anderson, *Ph.D., Computer Science and Engineering* Valuable Sheffey, *Undergraduate Student, Computer Science*

***Abstract*—With the United States' ever-increasing older population and a large demographic of people with functional diversity, it is more important than ever to ensure technology is universally accessible. Home automation systems have the ability to help bridge the digital divide and greatly enhance the lives of elders and those with disabilities. In this paper we present our vision of a platform-agnostic voice-operated (with a type option for those with vocal limitations) home automation app and the progress of the project so far. Our system involves a Windows 8.1 app built using Microsoft Speech API and a central home server built using openHAB software.**

***Keywords*—voice recognition, home automation system, smartphone app, elders, disabilities, digital divide, Microsoft Speech API, platform-agnostic, Window 8.1 phone**

I. Introduction

A disturbing trend of discrimination has been occurring amongst software developers and other designers of computer technologies. It is well-known that the US population of elders is increasing. In 2013, America held nearly 45 million people over the age of 64 [1]. The US Census Bureau predicts that by year 2033, for the first time ever, the population of elders in the US will outnumber children [2]. Not only do older individuals comprise a significant percentage of the United States population, but also do persons with special needs. In 2010, circa 56.7 million people in the US were recorded to have some kind of disability [3]. This already large figure is actually rather conservative when one considers it only takes in account noninstitutionalized civilians. As the report states, when including those in institutional group living, the population of those affected by disabilities increase appreciably. This data shows that elders and those with special needs form a considerable portion of United States citizens. Despite this however, very little effort has been made by designers to ensure that their technologies are accessible to such populations [4][5].

A digital divide, discrepancy between those with access to computer technologies and those without, exists notably for the elder and special needs population [5]. This is distressing because it is these excluded demographics who can benefit the most from such technology. Overcoming the digital divide is one step to ensuring more equality in access to opportunities and higher quality of life [5][4].

Dr.Anderson is with the Department of Computer Science, University of Alabama, Tuscaloosa, AL, 35487 USA e-mail: anderson@cs.ua.edu

Valuable Sheffey attends University of New Mexico

One possible piece to the solution is the creation of universal access tools. Such systems would integrate various technologies that are often difficult for elders and those with disabilities to use and merge interaction with these technologies into a single, approachable interface. For elders and those with disabilities, home automation systems can drastically improve quality of life by performing tasks that are physically-demanding. We propose that, when applied to home automation, the best implementation of afore-described universal access tool is a platform-independent voice-operated (with a type option for those with vocal limitations) phone app. This sort of app can help decrease the digital divide for elders and those with functional diversity and also provide them increased comfort and independence.

The aim of this paper is to present our progress on building such an app. This report is composed of four parts. The first will review previous work already completed on voice-operated wireless home automation systems and evaluate their pros and cons. The second will outline the intended design of our app. The third will report our progress thus far in implementing that vision. Finally, the fourth will present plans for future work.

II. Related Work

Several types of wireless home automation systems to assist the elderly and those with disabilities have been proposed.

In work by Alshueili, Gupta, and Mukhopadhyay [7], a wireless home automation system is presented that involves using a system of ZigBee devices controlled by the central control module, the PC. To perform a command one would speak into a hand-held microphone which was connected to the system via a ZigBee module. The pros of this system is that it is easy to install and configure. It is also voice-operated. A few cons though is that it is cumbersome to have to hold a microphone as one walks around the house. Also this system only works with ZigBee devices and therefore does not allow the freedom to mix and match devices from different vendors.

The project completed by Bittins, Sieck, and Herzog [6] provide some improvements compared to the described system above. Instead of a hand-held microphone, they utilize a smartphone. This is good because mobile phones are so ubiquitous in modern life that it would not be a barrier for the user to consider carrying one around the house. Their implementation uses KNX and has limited functionality due to the software required for such a system. Similar to above, this also limits the users' buying-freedom.

These works provide a foundation on which other researchers can build. They emphasize great features such as systems that are wireless (even better, one utilizes a smartphone) and voice-interactive. Ease of installation and configuration was also promoted. Where they both fall short, however, is in the restrictiveness of their systems. These days with so many types of devices on the market, home automation systems will do well to be platform-agnostic. We believe our approach employs the best of the above works while addressing the shortcomings. Our proposed app is platform-agnostic, voice-operated, easy to install and configure, and utilizes a smartphone.

III. Design

Our design for a platform-agnostic voice-operated home automation system involves three parts: the smartphone, the home devices, and the central control module. In this system, the smartphone is a device capable of handling voice commands and running the home automation app. The term "home devices" refers to automated appliances such as smart-fans and smart-doors. The central control module is a local server built using openHAB. This server is responsible for storing data about available devices and relaying information between the app and the devices. The app interprets given voice commands and sends these messages to the server. See Figure [1] for details.

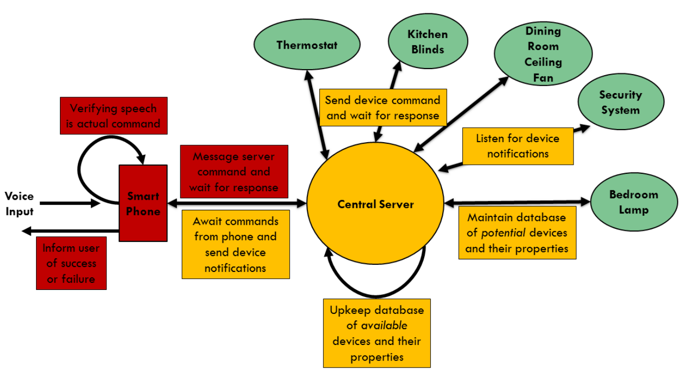

Fig. 1. Home Automation System Structure

This approach has many advantages such as:

A. Platform-Independence

The smart home market provides consumers with a diverse selection of vendors. The usual approach is to buy all the items from a single vendor due to the difficulty that would occur if one attempted to integrate devices that are without a universal communication method. However this closes consumers from sampling the diversity of devices, the optimal selection of which would likely employ a mix-and-match approach.

OpenHAB integrates various types of "home automation systems and technologies into one single solution that allows over-arching automation rules and that offers uniform user interfaces.[8]" In other words, openHAB allows the user to buy an assortment of home automation system technologies from different companies and interact with them all using openHAB's simple user interface. This platform-independence is a feature several approaches to home automation systems lack.

B. Natural Communication

Speech is human's natural form of communication. Therefore it is more intuitive than the usual type, click, and touch human-technology interactions. It is also the form of communication accessible to most levels of functional diversity (for those with vocal limitations text-interaction is still an option) and to people of all levels of technical literacy. For these reasons, voice is believed to be the optimal interface for a home automation system designed for elders and those with disabilities.

C. Mobile Application

In 2012 the Pew Research Center noted that 88% of Americans had cell phones [9] and found that 77% of the elder population owned a cell phone [10]. (No statistics for those with special needs were found). With the passing of three years, it is likely this number has increased. Such universality creates familiarity. A handheld or clip-on microphone would require a learning curve for the user to remember to carry, remove, and interact. On the other hand, the ubiquity of mobile devices makes their integration as a home automation device nearly imperceptible. Already accustomed to a cell, the only learning curve need be how to speak to the app. This and ease of updates makes mobile phones preferable as the medium for a home automation system.

D. "Skinny-App"

In consideration for mobile memory space, this design advocates for the creation of as "thin" an app as possible. The concept of "thin" refers to minimal processing by the app with most of the work performed by the home server. To accomplish this, the app's responsibilities were limited to voice-recognition, command interpretation, and communication with the server. The server's responsibilities however encompassed maintaining a comprehensive database of potential device types and their related commands, storing actual available devices and their properties, and relaying information between the app and many home devices. This separation of tasks allow for optimal processing and allowed the creation of an app of non-intimidating size.

IV. Implementation

The Process

Implementing the design involved coding the app, building the home server's database, and establishing communication protocols between the app and server. The Windows 8.1 phone app was built using Microsoft's Speech API and Visual Studio Community 2013. Constructing the server involved use of openHAB software. Basic networking was established fairly easily. Communication between the app and server used a RESTful API while openHAB's tools supplied server-device interaction. The research team split into two groups so that one would work on developing the home server while another would work on coding the mobile app simultaneously.

In the process of app-building, an unexpected feature of Microsoft's Speech API was discovered. The program allows for the customization of separate conversation definitions for interaction inside and outside the app. Interaction outside the app occurs with the Cortana interface. This interface provides the ability for the user to trigger and interact with the app without requiring a manual launch. This was pleasant surprise since other platforms (specifically Android and iOS) do not yet offer this feature in their development kits.

See Figure [2] and Figure [3] for a look at the conversation files.

```
<Command Name="SwitchLightON">
  <Example> Turn on Living Room Light</Example>

  <ListenFor> Turn [the] {switchdevice} on</ListenFor>
  <ListenFor> Turn on [the] {switchdevice} </ListenFor>

  <Feedback> Turning on the {switchdevice} ... </Feedback>

  <Navigate />
</Command>
```

Fig. 2. Part of VCD file for conversations with Cortana

```
<rule id="ON_FIRSTcommands">

  <example> Turn on Bedroom Light </example>

  <item> Turn on </item>
  <item repeat="0-1"> the </item>

  <!-- Programmatically developed list -->
  <ruleref uri="#DimDevice" />

</rule>
```

Fig. 3. Part of SRGS file for conversations within the app

However, it was found that certain limitations required by the Cortana interface (such as number of PhraseLists) may actually affect the build of the home server structure. More on this is discussed below.

After assigning groups, the team established a standard set of voice commands for the home automation system. At this point the project became more interesting. As the app group coded these commands into the program, the server group worked on coding the server to respond to these commands. This is where both groups discovered server set-up would be the most important portion of this project. Each time a new server framework was placed, new changes became necessary on the app side. After a few rounds of this, app work was placed on pause and further research focused on server structure.

This is the current point of the project. There are completed though temporary conversation files. The apps are able to successfully query the server for info and receive a response back. The home server is capable of receiving app information and interacting with the devices.

The Challenges

The two biggest challenges thus far has been (1) organizing the server database to be comprehensive, yet flexible and (2) determining messaging standards. To determine messaging standards means to decide on the format of queries to and responses from the server. Like the other parts of the project even this depends ultimately on (1) organizing the server structure.

In fact, server structure has proven to be so essential to this project that it deserves special elaboration. The expectations for the server was that it would be able to recognize a device connected to the home automation system and classify it according to its type. Device type would include elements such as whether it was a switch/dimmer/or other, what groups it belongs to, its name, and its aliases. See Figure [4] and [5]. From this comprehensive list of device types the researchers would create the conversation files and determine app-server message formats. However classifying the diverse assortment of devices that are available has proven more difficult than expected.

Consider an automated fan. Such a device may have three settings low, medium, and high. However, what if a particular vendor sells a fan that has four settings or even only two? Can a developing team really be expected to anticipate every setting type possible for a device?

A similar problem exists for devices that certainly belong in the same group yet lack not just settings, but full properties. Take for example doors. An automated garage door may imitate a regular garage door and only have a command function (meaning the door will open if closed, close if open). However some automated doors may have states, such as open or close, and specific commands about these states (door will only open if commanded open, only close if commanded close). Yet these are still all doors and, in some cases, these different properties can account for the same type of door (i.e. garage doors can be both stateless and stateful).

There is another server complexity unrelated to device grouping: the parameter issue. The typical temperature control system in the US allows for the choice between heat, cool, and fan. Therefore, setting the temperature involves at least two parameters (heat, 70; cool, 68; fan, auto; etc.). However openHAB is set up to receive only one parameter. This too must be eventually resolved.

Another consideration for the server is the format of the VCD file (file that outlines conversation definitions for interaction with Cortana). The VCD file has size limits on some of its elements. For example, PhraseLists may include no more than 10 items. If one were to use a PhraseList to represent something like device type, available slots may run out before all devices are listed. This effects what type of information the server should give the app to create the VCD file. Because of this, the logic of the organization of the VCD file will influence home server design.

The overall challenge is to figure out how to build a comprehensive database that encompasses all types of devices while avoiding locking the home into an inflexible structure. Potential solution approaches include: top-down (build an all-inclusive device type database and delete unnecessary options as devices area added) or bottom-up (starting from a near blank database, build device type files as each device is added), or perhaps some mix of the two.

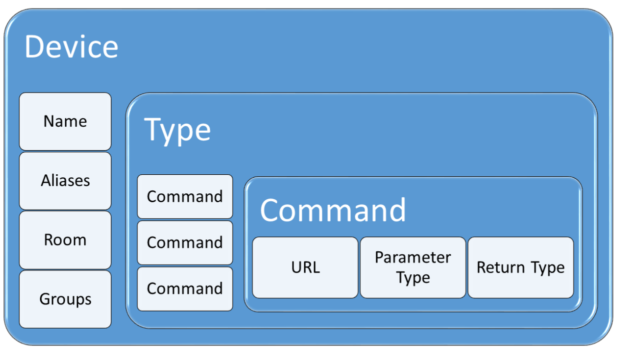

Fig. 4. Example Generic Device Framework

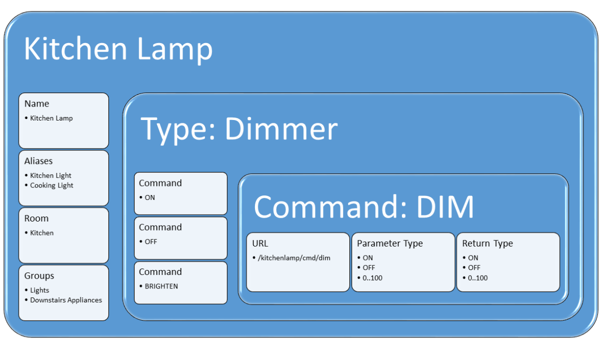

Fig. 5. Example Specific Device Framework

V. Analysis

This project was an attempt to build a platform-independent voice-operated home automation app that would improve the lives of elders and those with functional diversity. To accomplish this we choose to use Window's Speech API and openHAB software to create a prototype that can run on a Windows 8.1 phone.

A. Future Work

1) Notifications: A desired feature is that the app will alert the user whenever a remarkable event occurs (front door opens without command, alarm system manually disabled, etc.). Adding a mechanism for the server to communicate unsolicited events to the app should greatly increase the usefulness of the home automation system.

2) Friendly Avatar: A friendly-appearing avatar may lend concreteness to the user's interaction with the home automation system.

3) Audience Input: In-person interviews with the target audience (elders and those with disabilities), in their own homes preferably, should be performed if possible. This can allow researchers to get more insight about user desires and needs. As Soraghan found in his 2013 study, when engineers step outside their labs and interact with their intended audience, harmful misconceptions are destroyed and valuable information gained that can improve the user-technology experience [11]. Of course, user-testing a completed prototype at the end of a project is always recommended.

VI. Conclusion

Elders and people with functional diversity form a significant percentage of our population yet are often neglected when it comes to technology access. This is incredible considering technology has the potential to improve their lives the most. By building a platform-independent voice-operated home automation app, we hope to fulfill this need and lessen somewhat the digital divide. By combining the best of previous systems, we intend to build a more accessible system for these people.

Acknowledgment

Special thanks to Distributed Research Experiences for Undergraduates (DREU) for making this research possible.

References

[1] Census.gov, 'FFF: Older Americans Month: May 2015', 2015. [Online]. Available: https://www.census.gov/newsroom/facts-for-features/2015/cb15-ff09.html. [Accessed: 22- Jul- 2015].

[2] Census.gov, '2014 National Population Projections: Summary Tables', 2015. [Online]. Available: https://www.census.gov/population/projections/data/national/2014/summarytables.html. [Accessed: 22- Jul- 2015].

[3] M. Brault, Americans With Disabilities: 2010, 1st ed. US Census Bureau, 2012, pp. 70-131.

[4] H. Wandke, M. Sengpiel, and M. Snksen, Myths about older peoples use of information and communication technology, Gerontology, vol. 58, no. 6. pp. 564570, 2012.

[5] E. Gomez-Martinez and A. Fernandez-Diaz, Embedded Systems for People with Special Needs: Insights from a Real Case, 2014 40th EUROMICRO Conf. Softw. Eng. Adv. Appl., pp. 431436, 2014.

[6] B. Bittins, J. Sieck, and M. Herzog, Supervision and regulation of home automation systems with smartphones, Proc. - UKSim 4th Eur. Model. Symp. Comput. Model. Simulation, EMS2010, pp. 444448, 2010.

[7] H. AlShueili, G. Sen Gupta, and S. Mukhopadhyay, Voice recognition based wireless home automation system, 2011 4th Int. Conf. Mechatronics, no. May, pp. 16, 2011.

[8] openHAB.org, 2015. [Online]. Available: http://www.openHAB.org/features/introduction.html. [Accessed: 22-Jul- 2015].

[9] K. Zickuhr and A. Smith, 'Digital differences', Pew Research Center: Internet, Science & Tech, 2012. [Online]. Available: http://www.pewinternet.org/2012/04/13/digital-differences/. [Accessed: 23- Jul- 2015].

[10] A. Smith, 'Older Adults and Technology Use', Pew Research Center: Internet, Science & Tech, 2014. [Online]. Available: http://www.pewinternet.org/2014/04/03/older-adults-and-technology-use/. [Accessed: 23- Jul- 2015].

[11] C. Soraghan, S. Hermann, and G. Boyle, Design and older peoples real issues: Experiences of an engineer assessing technology in the user's home, 2013 6th Int. Conf. Hum. Syst. Interact. HSI 2013, pp. 600607, 2013.