Learnability through Adaptive Discovery Tools in Voice User Interfaces

Anushay Furqan
Digital Media
Drexel University
Philadelphia, PA 19104, USA
anushay.furqan@gmail.com

Chelsea Myers
Digital Media
Drexel University
Philadelphia, PA 19104, USA
cmm469@drexel.edu

Jichen Zhu
Digital Media
Drexel University
Philadelphia, PA 19104, USA
Jichen.zhu@gmail.com

CHI'17 Extended Abstracts, May 06-11, 2017, Denver, CO, USA
ACM 978-1-4503-4656-6/17/05
http://dx.doi.org/10.1145/3027063.3053166

Abstract
The invisible nature of VUIs has been attributed to challenging discoverability with VUIs. Low discoverability often leads to learnability issues. Researchers have designed visual tools for VUIs to help users learn as they go. However, few have used adaptation to ensure that learnability with the help of these tools extends beyond initial use. We designed *DiscoverCal*, a calendar application designed using adaptive discovery tools to improve learnability in VUIs. In this paper, we identify key characteristics of existing discovery tools. We present our design of a VUI that adapts based on contextual relevance and user performance in order to extend learnability beyond initial use. We briefly discuss our user study design.

Author Keywords
Voice User Interface; Learnability; Discoverability; Adaptive Interface

ACM Classification Keywords
H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous; H.5.2 [User Interfaces]: Voice I/O

Introduction
The way we interact with technology is evolving rapidly, making room for new interaction modalities beyond the keyboard and mouse. Voice User Interfaces (VUIs) have

matured significantly in recent years, and are now built into many of the devices we interact with on a daily basis from smart phones to televisions and cars. However, despite these recent advancements, the mainstream use of VUIs remains limited. A recent study found that even though 98% of iPhone users have tested Siri, 70% of these surveyed users continued to use Siri only "sometimes" or "rarely" [13].

The "invisible" nature of VUIs can challenge users' ability to discover its capabilities and limitations [7, 11]. When discoverability is challenging, learnability can be compromised. Learnability can be described as a novice users' ability to easily learn how to use a new system to maximum productivity, without any prior training [7, 8]. Different methods have been applied to improve discovery with commercially available VUIs, including tutorials, companion apps, or documentation of user manuals. Some researchers have designed tools to help users learn as they go [7, 9, 17]. However, few have studied how adaptation with these tools can effect learnability beyond initial use.

Our goal is to design visual Adaptive Discovery Tools (ADT) for a voice controlled calendar, *DiscoverCal*, in order to improve learnability in VUIs. ADT are guides that help users learn activities and commands that they have not yet discovered. We are using API.AI for voice input and output, and a wall mounted display for visual feedback. API.AI is a platform used to develop natural language interactions like Facebook chatbots and Amazon Alexa skills. We focus on building voice interactions with API.AI for this project, although we believe our findings can be generalized to a wide range of similar VUIs. API.AI has been chosen due to its open source nature as well as ease of accessibility for this project. Based on user performance, the discovery tools adapt content and level of visibility. The goal of ADT is to provide users with contextually relevant help during interactions, and to adapt with improved user performance, in order to learn to use the system to maximum productivity.

This paper provides a summary of our design approach to address some of the challenges with learnability in VUIs. A related work section highlights the state of the art, followed by a section identifying the characteristics found in related work that can inform the process of designing for discoverability. We then discuss how we have incorporated an adaptive approach in order to improve learnability. Lastly, we present our study design and plans for future work.

Related Work

For novice users to learn how to use a new system, it is important for them to first discover its capabilities and limitations. Discoverability is a means to achieve learnability [6, 7]. Tovi Grossman states that learnability is of two types, initial learnability and extended learnability [8]. Initial learnability is a novice user's ability to perform well during an initial task, whereas extended learnability is their ability to perform well or improve performance over intervals [8].

Tutorials are the most common approach applied to improve discoverability in VUIs [10]. However, they rely on a user to retain important information, and it can be cumbersome to retrieve it later on. Some commercial VUIs use a different approach. For example, Amazon's Alexa has a visual companion app that lists "Things to Try" which include sub-menus like "What's New" and "Discover Music". While beneficial for initial use, as a user becomes familiar with the technology, the basic menu options become redundant, and the information users need immediately is buried under layers of navigation. Companies like Amazon and Google are constantly adding new features compatible with their VUIs, making it important for users to be able to easily access information as they go, as well as discover more

advanced features to extend learnability. In the rest of this section we focus on two approaches for improving learnability in VUIs:

1. Learn As You Go Approach: Making information discoverable for users as and when they need it, so they can gradually develop a mental model of the interface [7].
2. Adaptive Approach: Adapting an interface to the needs of a user in order to extend learning beyond initial use [2, 5].

Learn As You Go Approach

The learn as you go approach emphasizes the need for interface tools that assist with discovery during interaction. A number of projects have operationalized this approach by designing what we will refer to as discovery tools. For example *ALADIN*, an assistive VUI for home automation, combines a tablet application that uses visual tools to inform users what devices can be controlled and the interactions available with each device [9]. In another study, *JustSpeak*, a universal voice control system for Android, makes context specific commands discoverable through labels on top of each actionable object, facilitating discovery on the go [17]. *VoiceNavigator* is a Mobile VUI utilizing tools such as "Discovery Prompts" to highlight new discoveries and a "What Can I Say" menu to assist users when they are unsure of what to say [7].

While these VUI projects apply a learn as you go approach using discovery tools, they utilize an additional input modality, namely touch, in order to cater specifically to users with accessibility needs. However, in the ubiquitous use of VUIs, providing an additional input modality can risk overcompensating for its limitations. An alternative source of input may cause users to use the more familiar touch modality, thus inhibiting learnability with VUI. Using a mobile or tablet also inhibits the benefit of VUIs as a hands-free interface. Our project intends to use discovery tools for the ubiquitous use of VUIs using a wall mounted display, where the display only serves to provide visual feedback.

Like the Alexa App, *ALADIN*[9], *JustSpeak*[17] and *VoiceNavigator*[7] largely focus on initial learnability. Repetitive information in discovery tools, as in the Alexa App, can get redundant and may even frustrate users [1]. Presenting the additional information required to extend learning can be irrelevant to some, and may clutter the interface. An adaptive approach can be utilized to personalize discovery tools to user needs.

Adaptive Approach

Adaptive Interfaces are designed to adapt their behavior to the needs of a user [2, 5]. Similarly, ADT can be designed to adapt to a user's level of expertise and context of use.

Interface adaptation is inferred based on triggers. User response times, and hastiness have been used as triggers to adapt playback speeds or verbal content in VUIs [15]. The same triggers have also been used with a mobile device to produce visual and verbal feedback about the VUI or target domain [12]. In both instances, interaction was quicker and users reported greater satisfaction when adapted feedback was relative to their skill level [15, 12]. Some patterns applied in Adaptive Graphical User Interfaces (GUIs) can be extended to the design of ADT, such as reducing visualizations of elements that are not necessary beyond initial use [16]. Other patterns include adapting content and prompts to frequency of use.

In most instances, adaptation with VUIs has been triggered by the quality of a user's speech. Using response time as a

trigger can be risky as a novice user's interruptions or hesitations may not be proportionate to their skill. Research suggests that adaptive interfaces should not give users a sense of reduced control [16].

From existing work we see that a number studies have operationalized a learn as you go approach to design tools to assist users with discoverability of VUIs. However, few have focused on an adaptive approach that encourages learnability to extend beyond initial use. We propose to design ADT for *DiscoverCal* that extend learnability by adapting to be contextually relevant and to a user's performance with the interface.

Designing for Discoverability in VUI

In order to improve the design for discoverability of VUIs, we surveyed existing projects [7, 9, 17] that utilized visual assistance to improve discoverability in VUIs and categorized the key design elements in them. These characteristics are summarized in Figure 1.

When designing for Discoverability in VUIs, it is important to make possible activities (what users intend to accomplish) and the commands (what users need to say to accomplish a task) to initiate them easily discoverable. This has been achieved through some of the characteristics embodied in the tools created for *ALADIN*, *JustSpeak* and *VoiceNavigator* [7, 9, 17]. For example, visualizing activities or commands is an important characteristic found in all these projects. In *ALADIN*, large buttons highlight activities, listing relevant commands within them. In *JustSpeak*, relevant commands are labelled on each actionable object. In *VoiceNavigator*, the "What Can I Say" menu reportedly improve learnability when the content was contextually relevant. Another characteristic in *VoiceNavigator* is visualizing new discoveries through "Discovery Prompts".

Characteristics	Tools in DiscoverCal
Visualizing what can be done	"What Can I Do?" menu
Visualizing what can be said	List of commands for what can be done
Visualizing New Discoveries	Discovery Prompts
Contextual Relevance	Adapting content + visuals

Figure 1: Characteristics identified from existing projects and corresponding Discovery tools for *DiscoverCal*

Unfortunately, users reported that Discovery Prompts fought for users' attention. The main reason for this is the limited screen space on the mobile device used for this project.

Visualizing what can be done (activities) and what can be said (commands) directly addresses the challenges faced by VUI as an invisible interface. Visualizing discoveries can give users a sense of confidence as they gradually navigate through the interface. Unlike GUI navigation, where a single command results in a single action, VUIs accommodate a number of commands for the same actions. Visualizing discoveries gives users a visual to track their progress. In the implementation of a learn as you go approach, contextual relevance is significant to ensure that learning content reflects a user's current context.

We identify these characteristics to inform the design of our project. However, we find that they can be used as general principles to design for discoverability in VUIs. *VoiceNavigator* successfully applied and tested some of these tools for learnability. However, they do not address extended learnability. We design *DiscoverCal* to extend this project by recreating some of those tools (List of commands and Discovery Prompts) and implementing additional tools based on the identified characteristics. Figure 1 lists the tools used in *DiscoverCal*.

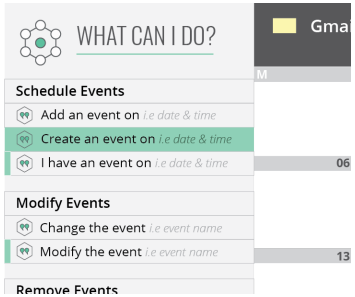

Figure 3: "What Can I Do" menu and listed relevant commands in *DiscoverCal*

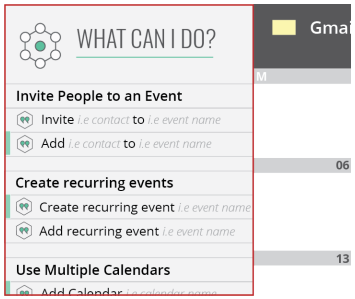

Figure 4: Adapting the menu and list to frequency of use in *DiscoverCal*

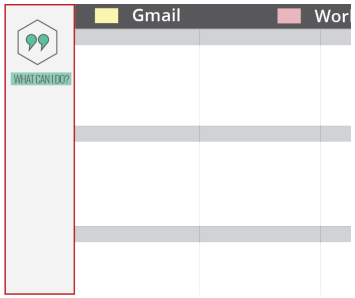

Figure 5: Adapting the visual interface in *DiscoverCal*

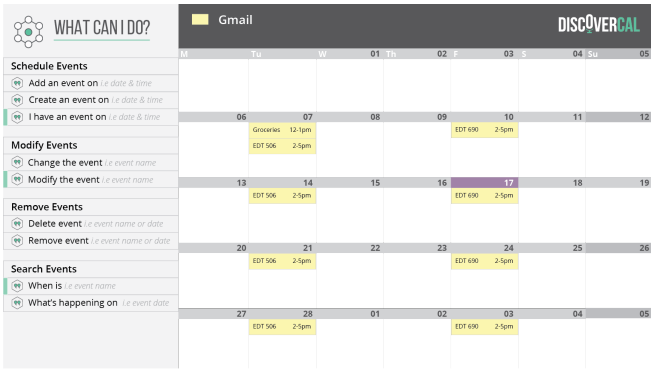

Figure 2: Screenshot of *DiscoverCal*

DiscoverCal

DiscoverCal (Figure 2) is a calendar system with both voice and visual interface. The user uses voice control to manage their schedule and receives visual feedback on a screen. In particular, we use API.AI as our VUI. The screen is designed to be any generic computer or TV screen. The focus is to explore how to design ADT to increase initial and extended learnability.

The aim of this research is to investigate how adapting discovery tools can increase learnability in Voice User Interfaces. We test the tools designed for this project (Second column in Figure 1) with a calendar, as it provides sufficient complexity to introduce adaptation and extend learnability over intervals. We chose the home as the context of use. A recent study found that 39% of users use voice interfaces at home while only 6% use them in public [13]. Task management and home automation have recently featured a popular application domain for VUIs like Amazon's Alexa and Google Home, but no existing systems has ADTs.

Once *DiscoverCal* is set up, users can begin learning basic activities and commands that are made visibly discoverable through ADT. After a command or activity has been successfully used, the content in the menu adapts. Adaptation is triggered by (a) user performance and (b) contextual relevance. Depending on how each individual user interacts with the system, ADT suggest activities and commands. For example (Figure 3), a user that frequently creates events is recommended content that advances that activity. Once the system detects a user has successfully performed that activity thrice, the system marks it as "learned" and removes it from the menu displaying it (Figure 4). Once users begin using the learned activities and commands that are no longer visualized in tools, the interface adapts to hide them (Figure 4 to 5), allowing users to pull them up with the displayed command if needed. In addition, it adapts the content of the tools to be relevant to that activity. For instance (Figure 6) content may adapted to recommend activities and corresponding commands relevant when creating an event.

ADT in *DiscoverCal* utilize a learn as you go approach to assist users in discovering features that are relevant to their current activity, and extend their learning of what the VUI is capable of. The characteristics identified in designing for discoverability have been operationalized through an adaptive approach to use discovery as a means to achieve learnability. Once users have successfully learned how to use the VUI, *DiscoverCal* serves to supplement to the interaction, as opposed to guiding it. The following persona and use case are demonstrate a typical user's interaction with the system.

Persona and Use Case

Lisa is a single mother working part-time while pursuing her MBA. She is often doing chores at home when she

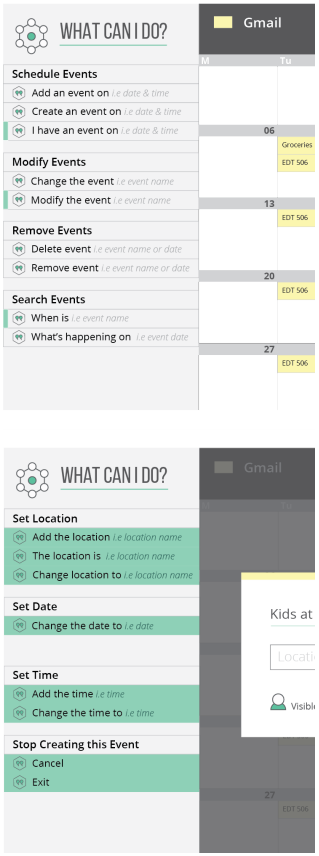

Figure 6: Adapting the "What Can I Say" and "What Can I Do" menu to be contextually relevant in *DiscoverCal*

remembers to update her calendar, so when she hears about *DiscoverCal* she decides to order it.

On Monday, Lisa sets up *DiscoverCal* on a wall mounted display in her living room. She quickly beings learning the basic tasks of adding reminders and events using the listed commands for what can be done with *DiscoverCal*. After she has successfully completed an activity and it's commands a few times the menu adapts to display more advanced options. For example, it adapts from "Schedule events" to "Use Multiple Calendars". Discovery prompts mark undiscovered features and briefly highlight a command as she discovers it, increasing her confidence.

By the end of the week, Lisa finds herself comfortable with the activities in the "What Can I Do menu" and is learning new commands to execute within them. On multiple occasions recently, Lisa has asked *DiscoverCal* to invite contacts to individual events. On Saturday, she notices an undiscovered activity, "Add people to events" in the What Can I Do menu and thinks, "I can share this with mom so she doesn't forget which weekend the kids will spend with her." She sees the appropriate command listed to share the calendar, "Share calendar with mom".

In the following weeks, Lisa continues to interact with *DiscoverCal*, learning new activities and commands with the help of ADT.

User Study Design

After iterating our design with formative assessment feedback we plan to evaluate our approach through a user study. *DiscoverCal* is designed for a wide range of users. Novice users of VUIs, who use VUI less than once a month, over 18 years of age will be recruited. Evaluation will be a between-subjects design that compares static discovery tools, which remain the same throughout interaction, with ADT designed for *DiscoverCal*. Participants will be randomly assigned to a control or test group (n=5 for each group [14]), and will be invited to complete a set of tasks with *DiscoverCal*. Participants will be tested over three 30-minute intervals, over a ten day period. Tasks will advance over intervals from basic activities such as creating events, to more complex ones such as adding a new calendar.

Task metric analysis and a System Usability Scale (SUS) questionnaire will be used to provide a statistical analysis of learnability in both user groups. Task metrics are used to store data on successful task completion and time taken to do so. The SUS is a ten-item scale that provides subjective quantitative user feedback about the overall usability of the system over time[3]. Observations and semi structured interviews will be used to provide a thematic analysis to assess learnability [4]. By comparing the data between the two groups of novice users over time, we can evaluate their learnability with the system[8, 3].

Conclusion and Future Work

We have presented our approach to address the challenge of improving discoverability in VUIs in order to extend learnability beyond initial use. After examining existing projects, we develop a set of characteristics that can contribute towards the improvement of discoverability in VUIs. We further extend the design of the tools created for *VoiceNavigator* [7], by designing and implementing an adaptive approach through ADT for *DiscoverCal* in order to extend learnability beyond initial use. A user study presents our future plans to provide an empirical evaluation of our design.

References

[1] Ken Abbott. 2002. *VUI Design Principles and*

Techniques. Apress, Berkeley, CA, 87–103.

[2] Pierre A. Akiki, Arosha K. Bandara, and Yijun Yu. 2014. Adaptive Model-Driven User Interface Development Systems. *ACM Comput. Surv.* 47, 1, Article 9 (May 2014), 33 pages.

[3] Jonatan Andersson. 2016. Assessment and Improvement of Initial Learnability in Complex Systems: A Qualitative Study to Promote Intuitive Software Development. (2016).

[4] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.

[5] Dermot Browne. 2016. *Adaptive User Interfaces*. Elsevier Science.

[6] Fang Chen. 2006. *Designing Human Interface in Speech Technology*. Springer US.

[7] Eric Corbett and Astrid Weber. 2016. What Can I Say?: Addressing User Experience Challenges of a Mobile Voice User Interface for Accessibility. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '16)*. ACM, New York, NY, USA, 72–82.

[8] Tovi Grossman, George Fitzmaurice, and Ramtin Attar. 2009. A Survey of Software Learnability: Metrics, Methodologies and Guidelines. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 649–658.

[9] Jonathan Huyghe, Jan Derboven, and Dirk De Grooff. 2014. ALADIN: Demo of a Multimodal Adaptive Voice Interface. In *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational (NordiCHI '14)*. ACM, New York, NY, USA, 1035–1038.

[10] Candace A Kamm, Diane J Litman, and Marilyn A Walker. 1998. From novice to expert: the effect of tutorials on user expertise with spoken dialogue systems.. In *ICSLP*. Citeseer.

[11] Laura Klein. 2015. *Design for Voice Interfaces*. O'Reilly Media, Sebastopol, CA.

[12] Kazunori Komatani, Shinichi Ueno, Tatsuya Kawahara, and Hiroshi G. Okuno. 2005. User Modeling in Spoken Dialogue Systems to Generate Flexible Guidance. *User Modeling and User-Adapted Interaction* 15, 1 (2005), 169–183.

[13] Carolina Milanesi. 2016. Voice Assistant Anyone? Yes please, but not in public! (June 2016). http://creativestrategies.com/voice-assistant-anyone-yes-please-but-not-in-public/

[14] Jakob Nielsen. 2000. Why You Only Need to Test with 5 Users. (March 2000). https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/

[15] Daniel O'Sullivan. 2009. *Using an Adaptive Voice User Interface to Gain Efficiencies in Automated Calls*. Technical Report. Contact Solutions | Inventing Real Customer Service, Smithtown, USA.

[16] Qian Yang, John Zimmerman, Aaron Steinfeld, and Anthony Tomasic. 2016. Planning Adaptive Mobile Experiences When Wireframing. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems (DIS '16)*. ACM, New York, NY, USA, 565–576.

[17] Yu Zhong, T. V. Raman, Casey Burkhardt, Fadi Biadsy, and Jeffrey P. Bigham. 2014. JustSpeak: Enabling Universal Voice Control on Android. In *Proceedings of the 11th Web for All Conference (W4A '14)*. ACM, New York, NY, USA, Article 36, 4 pages.