ALADIN: Adaptive Speech Interaction for People with Disabilities

Jonathan Huyghe
CUO|Social Spaces, iMinds
KU Leuven
Leuven, Belgium
jonathan.huyghe@soc.kuleuven.be

David Geerts
CUO|Social Spaces, iMinds
KU Leuven
Leuven, Belgium
david.geerts@soc.kuleuven.be

Jan Derboven
CUO|Social Spaces, iMinds
KU Leuven
Leuven, Belgium
jan.derboven@soc.kuleuven.be

Dirk De Grooff
CUO|Social Spaces, iMinds
KU Leuven
Leuven, Belgium
dirk.degrooff@soc.kuleuven.be

Abstract
This position paper gives an overview of our ongoing work within the ALADIN project, which aims to develop an assistive vocal interface for people with physical impairments. Unlike most current Automatic Speech Recognition solutions, the system is entirely trained by the user, which provides extra challenges to the design of the interface. We describe three iterations of our user tests, showing how constraints and multimodal design influence the user expectations and interactions.

Author Keywords
Assistive technology; Speech interaction; Self-learning systems

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

Introduction
While there is a growing trend towards mainstream uses of Automatic Speech Recognition (ASR), there are some unique opportunities and challenges for niche applications of speech technology. One of these is the domain of assistive technology, where vocal commands can allow people with motor impairments to significantly simplify their daily tasks [4]. By gaining the ability to easily control their home, domestic

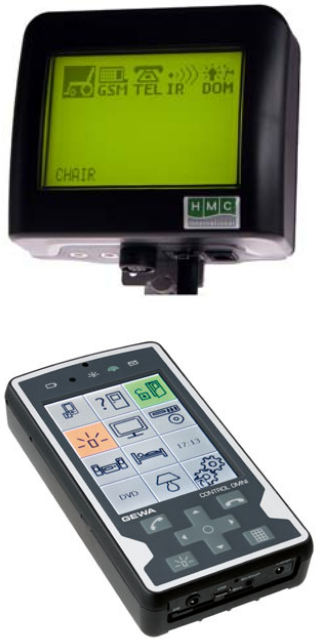

Figure 1. HMC "Easy Rider" (top) and "Gewa Control Omni" (bottom), two multifuntional interfaces for people with disabilities, operated through switches or scanning.

Figure 2. Sketched scenario used during our first tests.

appliances, or entertainment devices, voice control could contribute to their independence of living and quality of life. Current solutions rely mostly on button-based remote controls or a graphical user interface (see figure 1) operated using switches, which are controlled with varying ease-of-use.

In this paper, we describe a speech recognition system for people with disabilities developed in the ALADIN project. We describe the overall project goals and three iterations of user tests, focusing on how our methodology of testing influenced the way in which users interact with the vocal interface.

Aim of the project

The ALADIN project was set up to create an adaptive, learning speech recognition system for people with disabilities, offering control over a wide range of applications. So far, vocal interfaces have not yet seen a wide adoption in assistive technologies, despite the obvious advantages as an interface for people whose impairment restricts (upper) limb use and thus their ability to use more traditional remote controls. There are several reasons why speech recognition is difficult to implement for this target group:

- A lot of users who could benefit from voice control due to motor impairments also suffer from a speech pathology, making state-of-the-art speech recognisers unusable by them.
- Current vocal remotes require the user to use pre-defined commands, forcing them to adapt to the system and learn the proper commands.
- Progressive diseases often lead to changing speech patterns, which requires a constant adaption of the system.

There are already a number of solutions that address some of these problems, but are lacking in other aspects: the Pilot Pro [2], for example, offers a fixed number of pre-programmed functions and a very hard to use training method. Castle OS [1], a more recent solution that is not aimed specifically at people with disabilities, features a more intelligent and expandable set of controls, but uses natural language recognition, unsuitable for people with speech impairments.

The aim of the ALADIN project then, is to provide users with a system that can be adapted to their specific living situation and can learn their commands instead of the other way around, deducing grammar and vocabulary from the user's speech.

Design process

As this project is a collaboration between HCI researchers and speech recognition researchers, the work on both sides necessarily runs in parallel. We describe the users' interactions during early user testing, when the speech recognition engine was far from ready, even for rudimentary testing.

Sketched scenarios

In a first exploratory study, we gauged how people would want to use a voice-controlled home automation system using the 'sketched scenario' method, in which we presented users with visualisations of interactions, and asked them to utter the voice commands they would use to control this interaction (see figure 2).

The focus of this study was not to simulate system interaction in a very realistic way, but rather to explore the variation in how the targeted user group addresses a voice interaction system. Significant diversity was

found in interaction styles: voice commands ranged from a purely 'technical', command-style interaction to a more anthropomorphized, natural communication with the system. Furthermore, variation in commands ranged from addressing the system as a whole (telling the system to act on the environment) to addressing individual devices, without addressing the system as a whole. Although it is user-trained, the speech recognition system does need clear commands as input, including a clear starting and endpoint, to avoid unwanted actions. As the variation in this first test iteration would not be beneficial to the functioning of the system, We tried to anticipate this problem in the next test by narrowing down the interaction possibilities presented to users.

Wizard of Oz

The second, medium-fidelity, approach to user tests came in the form of Wizard of Oz testing, which has its roots in the testing of ASR applications [3]. Our main concerns here were the usability of the system and variation in commands used by participants. Because we mainly focused on home automation, we needed an efficient way of simulating typical home automation tasks during on-location tests with users, whose mobility was often limited. For this we created a virtual 3D environment using Unity 3D, modeled after an adapted home for people with disabilities (figure 3). We could open doors, turn on lights, adjust the bed, etc. from a separate interface, allowing a researcher behind to scenes to manipulate the 3D home based on voice commands from the user, who was taken through a scenario with a moderator. This proved to be a much more immersive experience for users, and created a more realistic representation of the envisioned interaction.

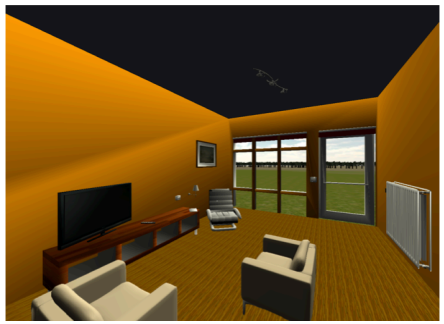

Figure 3. Screenshot from the 3D virtual home.

In the Wizard-of-Oz tests, participants were asked to address the system using a system name before uttering their commands. While this was necessary primarily for technical reasons, this change resulted in a smaller diversity of command styles: as participants had to name the speech system, they no longer addressed individual devices in the environment, but addressed the system as a whole. In other words, this primarily technical constraint limited the users' interaction styles, making the participants' commands more coherent. This meant it was easier to have a uniform starting word/phrase, which taken together with the smaller variation in commands, aligns better with the capabilities of the speech recognition system.

However, users continued to approach the voice control system in a way that is much more natural and conversational than the system is capable of, leading to (simulated) breakdowns in recognition. The main ways in which this happened were a reliance on spatial and conversational context and unsuccessful negotiation with the system. Reliance on context meant users formulated incomplete commands, such as "open the door", instead of specific commands such as "open the bedroom door". This type of command rests on the assumption that the system is able to understand the context of interaction: both the spatial context of the user, and the conversational context of devices that have just been referred to. Negotiation was a different problem that occurred when users added new requests after a command or tried to correct a wrongly formulated command, which would not be recognised using the real speech recognition system. Mitigating these problems will rely mostly on providing users a clearer picture of the system state, such as showing when exactly it is listening or what commands it has

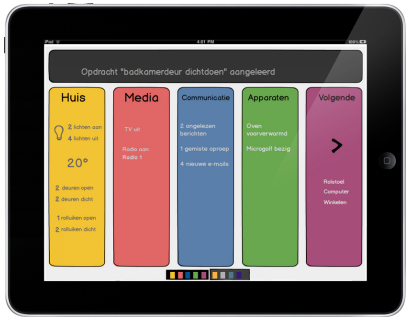

Figure 4. Mockup of the tablet application

recognised. This is one of the reasons to introduce (optional) multimodal interaction.

Multimodal interface

While the ALADIN system should be able to function as a stand-alone device, we developed a secondary touch-based tablet interface to provide extra functionality, improving the user experience. Its main functions are to (1) provide richer feedback from both the system (showing when the system is listening, or reporting possible problems) and the devices (showing which lights in the house are still on, the temperature of the thermostat, etc.), (2) function as a back-up interface for correcting misunderstood commands or as a fallback, and (3) provide an easier and centralised system for training the system.

While using this input method seemingly defeats the purpose of having voice control, we have adapted it to our user group by using large vertical buttons that can be activated using swabbing, which means a button is selected upon release of a finger input, rather than on the first contact, a method originally developed and successfully tested for older people with tremors [5]. Furthermore, the interface can be used by caregivers during the heaviest training period, or by users themselves using their existing scanning/switch inputs.

During the development of this tablet interface, we used an interactive mock-up of the application which could send and receive information from the 3D home used earlier. Because we did not yet have a functional speech recognition system, a second researcher controlled the application and 3D home from a separate interface imitating the speech engine, in a Wizard of Oz setup.

The extra information offered on the tablet interface limited the variation in commands even further. By seeing feedback about the system state, users also get information about devices that can be controlled, and which states are available. For instance, in a home automation environment, users get feedback on which lights they can control, how they can address them, and which states are available (e.g. different brightness levels for dimmable lamps vs. binary on/off for non-dimmable lamps).

Conclusion

Throughout our different iterations in prototypes and methodology, we have tried to bring the user interaction closer to the technical possibilities and limitations. We have achieved this by progressively making the interface and interaction more concrete, adding reactivity (Wizard of Oz method) and offering users more information (Multimodal interface).

References and Citations

[1] CastleOS Software, LLC. CastleOS – Home automation made simple, Retrieved January 10: http://www.castleos.com

[2] NanoPac, Inc. Sicare Pilot Pro – Control Your Environment, Retrieved January 10: http://www.nanopac.com/SiCare%20Pilot%20Pro.htm

[3] John D. Gould, John Conti, and Todd Hovanyecz. 1983. Composing letters with a simulated listening typewriter. *Commun. ACM* 26, 4 (April 1983), 295-308.

[4] Noyes, J. and Frankish, C. Speech recognition technology for individuals with disabilities. *Augmentative and Alternative Communication 8*, 4 (1992), 297–303.

[5] Wacharamanotham, C. et al. 2011. Evaluating swabbing. *Proc. CHI '11,* (2011), 623.