

Spring
2014

Engaging Speech UI's

HOW TO ADDRESS A SPEECH RECOGNITION INTERFACE

15P MASTER THESIS - COMPUTER SCIENCE

AUTHOR

Hampus Söderberg
hampus.soderberg@gmx.com

SUPERVISOR

Bo Peterson
bo.peterson@mah.se

EXAMINER

Edward S. Blurock
edward.blurock@mah.se



MALMÖ UNIVERSITY
Faculty of Technology and Society
Department of Computer Science

Abstract

Speech recognition has existed for a long time in various shapes, often used for recognizing commands, performing text-to-speech transcription or a mix of the two. This thesis investigates how the input affordances for such speech based interactions should be designed to enable intuitive engagement in a multimodal user interface.

At the time of writing, current efforts in user interface design typically revolves around the established desktop metaphor where vision is the primary sense. Since speech recognition is based on the sense of hearing, previous work related to GUI design cannot be applied directly to a speech interface. Similar to how traditional GUI's have evolved to embrace the desktop metaphor and matured into supporting modern touch based experiences, speech interaction needs to undergo a similar evolutionary process before designers can begin to understand its inherent characteristics and make informed assumptions about appropriate interaction mechanics.

In order to investigate interface addressability and affordance accessibility, a prototype speech interface for a Windows 8 tablet PC was created. The prototype extended Windows 8's modern touch optimized interface with speech interaction. The thesis' outcome is based on a user centered evaluation of the aforementioned prototype.

The outcome consists of additional knowledge surrounding foundational interaction mechanics regarding the matter of addressing and engaging a speech interface. These mechanics are important key aspects to consider when developing full featured speech recognition interfaces. This thesis aims to provide a first stepping stone towards understanding how speech interfaces should be designed.

Additionally, the thesis' has also investigated related interaction aspects such as required feedback and considerations when designing a multimodal user interface that includes touch and speech input methods. It has also been identified that a speech transcription or dictating interface needs more interaction mechanics than its inherent start and stop to become usable and useful.

Tags

speech recognition, interaction design, user experience design, HCI, NUI, multimodality

Popular science summary

Speech recognition technology allows the conversion of analog human speech into digital text strings that a computer can understand and manipulate. This process is commonly known as *speech-to-text transcription*. The technology itself has existed for a long time and several commercial and academic projects have helped develop the technology to the point of being practically usable. The projects have predominantly been done from a technical development perspective meaning that the focus have been to develop accurate, reliable, fast and efficient speech recognition systems. Typically, these projects have had a spoken or unspoken intention of creating speech as a replacement or complement to existing input methods such as touch or mouse and keyboard.

The work in this project was done from another perspective, namely an interaction designer's. *Interaction design* is a field that investigates how products should look, behave and most importantly *feel* when it is being used by people. This makes interaction design a crucial part of the development of any product - because if people don't understand the product, the product is not found useful by those who needs or should want to use it and is thereby not usable. Likewise, if the product doesn't solve the problems it was originally created to solve, the product is also less than a success.

Regarding speech recognition, the technological underpinnings have been under active development for a long time and although there have been multiple commercial products available from a lot of different vendors, the technology is yet to be the mainstream. One could argue that it is the lack of accuracy or similar technical reason that is the cause for this. However, the author's inspiration for doing this investigation is a belief that now that the technology is ready, it is time to figure out how to make it useful and thereby usable. In order to make something useful, that something needs to bring a clear advantage that makes people wonder how they could've ever been able to cope without it. A great deal of time and effort is therefore invested in exploring potential usage areas for new technologies such as speech recognition. Something that is particularly important for technologies that are developed to enable new ways for people to interact with computers, is that the individualities of both the people and the new interaction method needs to be fully understood. If they are, it is possible to make a product that feels second-nature to the people using it.

The results of this thesis should provide a first stepping-stone towards understanding how speech recognition can be used to create products that are useful. In detail, the beginning of a speech interaction will be investigated to find a design that makes it feel natural to start talking to a computer.

The investigation concluded that by using the style and structure of an everyday human-to-human conversation, it is possible to make the experience of talking to a computer include certain aspects of a humanistic feel, traditionally only experienced when talking to people. It was also found that speech has the potential of complementing touch as an additional input method that can be used interchangeably without changing context.

Dictionary

- Calm TechnologyTechnology that is unobtrusive and exist in the periphery of a persons' attention field.
- Faceless.....A computer or digital device that lacks a graphical display.
- GUI.....Graphical User Interface.
- HCI.....Human-Computer Interaction.
- Hi-Fi prototypeA polished prototype that demonstrates a working implementation of a design. Might lack application logic.
- Input method.....Means with which you can interact with technology.
- Interaction modeAll input methods rely on different mechanics and offer different affordances, hence an interface need different interaction modes.
- ModelessBeing able to transition seamless between different interaction modes makes an interaction appear modeless.
- MultimodalityThe combination of several input methods.
- NUI.....Natural User Interface.
- OmnipresentBeing always readily available.
- Peripheral attentionBeing aware of something without giving it direct focus. Similar to peripheral vision.
- Scenario.....A fabricated setting that can be used to provide context for a concept or prototype.
- SkeuomorphismDigital design analogy to a non-digital artifact. Example: Providing a digital clock with the face of a wristwatch.
- Socially Natural.....A HCI experience that is logical and reminiscent of HHI.

Table index

Table 1: Method overview showing how the research methods will be applied..... 5

Illustration index

- Illustration 1: A voice activated Windows 8 UI without visual hints. 23
- Illustration 2: Voice activated Windows 8 UI with the "feedback ribbon". 24
- Illustration 3: Voice activated Windows 8 UI with "feedback toast". 24
- Illustration 4: Voice activated Windows 8 UI with "dynamic action". 25
- Illustration 5: Voice activated Windows 8 UI with "voice keyboard". 26
- Illustration 6: Voice activated Windows 8 UI using a toast to present a speech input modality.
..... 27
- Illustration 7: Revised version of the "dynamic action"-concept. 28
- Illustration 8: Revised version of the "voice keyboard"-concept. 29
- Illustration 9: Technical Proof of Concept using Microsoft Bing Speech..... 30
- Illustration 10: Technical Proof-of-Concept using Microsoft SAPI. 31
- Illustration 11: Speech interface components map..... 32
- Illustration 12: Default Windows 8 modal ribbon dialog. 33
- Illustration 13: Conceptual feedback ribbon. 33
- Illustration 14: Default "notification toast" used to notify user of a new e-mail message on
Windows 8. 33
- Illustration 15: Modified notification toasts which act as affordance presentation and
information output elements. 33
- Illustration 16: Default Bing speech transcription GUI. 34
- Illustration 17: Default Bing Speech transcription GUI "docked" to the default on-screen touch
keyboard. 34
- Illustration 18: Flow chart of the Hi-Fi prototype. 38
- Illustration 19: Verbal command model inspired by HHL..... 40
- Illustration 20: Hi-Fi prototype - Speech for launching. 41
- Illustration 21: Hi-Fi prototype - Speech for getting answers..... 42
- Illustration 22: Hi-Fi prototype - Speech for writing. 44

Table of contents

Abstract.....	i
Tags	i
Popular science summary	ii
Dictionary.....	iii
Table index.....	iv
Illustration index	v
1. Introduction.....	1
1.1 Problem area	2
1.2 Goals and purpose	3
1.3 Research questions	4
2. Method.....	5
2.1 Literature review	6
2.2 Sketching	7
2.3 Scenarios.....	8
2.4 Prototyping	8
2.5 User testing.....	9
3. Literature review	11
3.1 Background	12
3.1.1 Traditional GUI design	12
3.1.2 Human-human interaction	13
3.1.2.1 Verbal and non-verbal messages	13
3.1.2.2 Multimodality	14
3.1.4 Sonic feedback.....	15
3.2 Related work	16
3.2.1 Previous research	17
3.2.1.1 Accessibility.....	17
3.2.1.2 Transcribing speech - dictating instead of typing	17
3.2.2 Previous commercial work	18
3.2.2.1 Dictating with “Dragon”.....	18
3.2.2.2 Personal assistant “Siri”	19
3.2.2.3 Speech platform “SAPI”	19
3.2.2.4 Dictating with the “Input panel” in Windows XP	19

3.2.2.5	Fully voice enabled GUI in Windows Vista	20
3.2.2.6	Cloud based speech platform.....	20
3.2.2.7	Voice controlled “launcher” in Windows Phone 7	20
3.3	Conclusion	21
4.	Design process	22
4.1	Conceptualization	22
4.1.1	First round of concept sketching.....	23
4.1.2	Second round of concept sketching.....	26
4.2	Technical Proof-of-Concept.....	30
4.3	Hi-Fi prototype components and mechanics.....	32
4.3.1	Interface components	32
4.3.1.1	Feedback ribbon	32
4.3.1.2	Feedback toast.....	33
4.3.1.3	Feedback sounds.....	34
4.3.1.4	Speech transcription bar	34
4.3.2	Interaction mechanics	34
4.3.2.1	Addressing a device by name	35
4.3.2.2	Social command model	35
4.3.2.3	One-shot dictating.....	35
4.3.2.4	Context activated dictation.....	36
4.3.2.5	Feedback sounds.....	36
4.4	Hi-Fi prototype.....	36
4.4.1	Flow chart.....	37
4.4.2	Speech for launching	39
4.4.3	Speech for getting answers	42
4.4.4	Speech for writing.....	43
4.5	User test.....	45
5.	Summary and analysis	47
5.1	Summary	47
5.2	Speech for launching & getting answers	47
5.2.1	Presentation	48
5.2.2	Activation	48
5.2.3	Sonic feedback.....	48
5.2.4	Visual response	49

5.2.5	Analysis	49
5.3	Speech for Writing.....	50
5.3.1	Presentation	50
5.3.2	Activation	50
5.3.3	Sonic feedback.....	51
5.3.4	Visual response	51
5.3.5	Analysis	51
6.	Conclusion and discussion.....	53
6.1	Conclusion	53
6.2	Discussion	54
6.3	Future work	55
7.	References	56
	Appendix A: Test plan	59

1. Introduction

Speech recognition is a well-visited research area that has been investigated from different perspectives. Popular culture have envisioned speech to become an efficient input method that would ultimately replace keyboard and mouse interactions. Social studies have envisioned speech interfaces to have the potential of connecting more people by removing language barriers. Lastly, computer scientists have done a tremendous amount of work to enable the accurate and reliable experiences that are possible with today's speech recognition technologies.

Current human-computer interactions (HCI) are generally screen based. Since its conception in the mid 1970's, the desktop metaphor has played a key role in the development and design of HCI. Graphical user interfaces (GUI's) based on the desktop metaphor are at front and center in both Microsoft's Windows and Apple's OS X operating systems.

For the past seven years, the market for digital devices has evolved rapidly to embrace more natural interaction methods, most prominently touchscreen input. Touch is commonly regarded to have revolutionized HCI but there are still situations where a keyboard or a mouse is considered to provide a better experience.

Recent technical progress in developing fast and accurate speech recognition technologies is enabling new and possibly improved user experiences for HCI. Comparable to how the market adopted touch-enabled devices, device manufacturers and software developers have started to extend the touch-only GUI experiences on smartphones with speech recognition.

Current commercial solutions such as Apple's Siri, Microsoft's Cortana and Google Now, all share a common design trait - namely they all rely on vision to initiate an interaction. This is an incremental design approach that tries to apply existing design patterns to a new interaction method. While maintaining familiarity, it does so at the possible expense of a lost opportunity to innovate using the qualities that are inherent to the new interaction method. In this case, speech recognition is that other method which adds fundamentally different interaction capabilities. When compared to a mouse or touch operated GUI, a key difference is the communicative medium. GUI's rely on vision to present available interactions to the user, meaning that it is the computer that makes the first move in an "interaction dialog" by showing its user what the possible actions are. A speech interface on the other hand, relies on audio and could be seen as to work in reverse when compared to a GUI. This, since it would often not make sense for a speech interface to continuously speak out all the available speech interactions to the user, hence it is the user who should engage the interaction dialog and not the interface as is typically the case in a GUI.

In order for any interaction method to become useful, it is important to understand its inherent qualities. This thesis aims to increase the understanding of how speech interfaces should be designed to enable intuitive speech interactions. The belief is that by investigating how to engage a speech interface, an important milestone towards sensible and efficient speech interfaces can be reached.

1.1 Problem area

This master thesis in computer science focuses on Human-Computer interaction (HCI) in the context of a natural user interface (NUI). This field of research investigates how human-computer interaction can become more natural. It explores what future possibilities cutting-edge technologies may enable, if they are appropriately designed. It does not however investigate the technical aspects of designing a speech recognition engine. Rather it focuses on making this technology useful by exploring potential usage scenarios and investigating how an intuitive speech interface should be designed.

Currently, most efforts within the HCI field are put in graphical user interface (GUI) design which builds on the desktop analogy that was conceived in the mid 1970's. (Moggridge 2007) The idea of a desktop with clickable on-screen elements came from the notion that a computer is an office machine that handles documents. In an effort to simplify computer interaction, the computer received a GUI that resembled a real-world desktop, complete with movable documents and folders (Mott 2003).

Devices that are designed upon this paradigm, relies on a display with a GUI that presents visual content (i.e. text, pictures) and interaction affordances (i.e. buttons and menus) to users. This works well for office and media consumption devices such as a workstation PC's and smartphones because they are essentially made to display rich media content.

Workstation PC's are still used in offices to handle documents and smartphones are often used for taking and sharing pictures or reading news articles.

As indicated by Dan Saffer (2007), the movement towards *ubiquitous computing* will likely increase the general interest in *faceless interfaces*, such as speech recognition. Ubiquitous systems are distributed across many different environments and may not always have any display affordances available; it may not even require one.

Before train-ticket machines entered stage, train tickets were bought over a counter. The soon-to-be passenger *asked* the cashier for a ticket to a certain destination. The cashier *responded* by restating what he or she had heard and asked the customer if he or she was sure. At that point, the customer either confirms and receives the appropriate ticket or tells the cashier his or her destination a second time. This represents a very simple human to human interaction (HHI) that was completely ignored when today's touch and GUI based train-ticket machines were developed. As a result of this negligence, not all customers are able to buy a ticket, which is a disaster to all stakeholders; the customer is unable to receive the requested service, the train operator misses a business opportunity and might receive a bad reputation that could mean long term damage to their business. (Schreder, et al. 2009)

The ticket machine situation is an example of where a speech interface might be more suitable than a traditional GUI. Before speech interfaces can be widely adopted and used, researchers and designers need to identify their inherent strengths and investigate how these could be exploited.

Bellotti et al. points out that it is important to understand that knowledge about traditional GUI design cannot be applied directly to a NUI such as a speech interface because they are too fundamentally different (2002). While traditional point-and-click GUI's require very explicit user actions, the mouse pointer has to move to a specific location on the screen and has to be followed by either a left- or right-click to execute an action, HHI works very

differently. In addition to verbal messages, words, humans also make extensive use of non-verbal messages when communicating face-to-face. Non-verbal messages include, among others: posture, facial expressions and sounds. These are vital for a successful human-human interaction. (Morency, Modeling Human Communication Dynamics [Social Sciences] 2010) In a GUI, any available interactions are presented visually as buttons. This means that in contrast to speech UI's that work by audio, GUI's are visual by nature. The research in this thesis is done from a perspective of exploring how a speech may stay true to its inherent audial communicative medium by providing a speech interface that doesn't require or rely on visual components to drive an interaction.

1.2 Goals and purpose

The overall purpose of this thesis is to explore how speech recognition's inherent characteristic of being faceless can be exploited to enable new and potentially more natural and intuitive ways of interacting with computers. Furthermore, it sets out to explore how today's GUI based touchscreen interfaces can be augmented and extended with a speech input modality.

The particular goal is to investigate fundamental interaction mechanics needed for successful speech interactions in a HCI interface. In this sense "successful", means that the interface should be perceived as useful and intuitive by its users.

The focus will be on interaction mechanics related to the matter of addressing a speech interface. For a speech interface to be useful it is important that it knows when it is being addressed and when it is not. Two methods for addressing a speech interface will be evaluated; one that builds on HHI mechanics and one that is done from a perspective of multimodal input.

The investigation of the first method will start with a review of basic HHI topics in an attempt to better understand the mechanics of verbal human-human communication. The intention is to recognize HHI mechanics that could also be relevant for a speech interface and then incorporate those in a design proposal.

The investigation of the second method will start with a literature review that examines the topic of multimodality. The outcome of the literature review will be the foundation for a forthcoming design proposal.

To carry out this investigation, a speech interface will have to be created. Furthermore, for this interface to be suitable for evaluation, it needs to be approachable and make sense to user test participants. This will be solved by framing the speech affordance and its related speech interface inside contextual scenarios.

While the goal is to investigate how to engage a speech UI, it's expected to also learn about other interaction mechanics such as the kind of feedback and guidance that is required for a successful speech interaction. Additionally, some insight into when and how a speech interaction interface makes sense is also expected to come out of this investigation.

1.3 Research questions

1. *Using common HHI patterns, how can a speech interface be designed to be perceived as socially natural to engage a speech interaction with?*
2. *How can a speech dictation affordance be integrated into a multimodal touch and speech input interface to offer seamless activation?*

2. Method

This thesis comprises a qualitative study that consists of three stages: the process of gathering relevant information, the design process and the evaluation process. The method choices are motivated and inspired by Cooper, Reinman and Cronin (2007) who argue that qualitative methods are better suited than quantitative for usability practices.

Numbers are the outcome of any quantitative research method. These numbers may provide significant insight when used in hard science research, such as a study in physics. This is because, in such a study, it typically is possible to keep a strict control of all dependent and independent variables.

Conversely, for evaluating the perceived experience by a human user, the number of dependent variables increases drastically along with a number of unknown variables caused by unforeseen circumstances which are naturally occurring when dealing with people.

Because of this, methods that are original to social sciences such as anthropology have been embraced by interaction designers as tools for better understanding their users. (Cooper, Reinman and Cronin 2007)

Getting a thorough understanding of the problem area and knowledge of previous academic and commercial work is important to be able to make useful discoveries. A literature review is consequently the first work stage for this thesis project. The collected data is useful in the design process for which the gained knowledge provides a foundation.

The design process consists of common interaction design methods such as sketching, scenario creation, prototyping and user testing. While these methods are accumulative, with each part generating information needed for the next step, it is through sketching that a potential answer to the research questions will be formed. To make this potential answer testable, it will be turned into a working prototype which will be used for user testing. The user testing will generate insight into the perceived usability of the proposed design decision. In this sense, the user testing method will generate the data required for answering all of the research questions. See Table 1 below for details.

Gather knowledge	Design process	Result
Learn about previous research (Method: Literature review)	Identify use scenario (Method: Sketching)	Learn about perceived usability of prototype. (Method: User testing)
Learn about previous commercial solutions (Method: Literature review)	Define use scenario (Method: Sketching)	
	Create testable implementation. (Method: Prototyping)	
Identify design gap	Propose solution	Evaluate proposed solution

Table 1: Method overview showing how the research methods will be applied.

The motivation for selecting these research methods and rejecting others, was based on past experiences and the selected methods' inherent qualities of being user centric even though there currently are no existing users. In this project, the sketching method is based on the outcome of the literature review. In an effort towards generating high quality sketches, it was deemed necessary to examine successful and unsuccessful designs of the past to make grounded design decisions. Another approach could be to do anthropological studies such as interviews to gather information about required features and user expectations. However, because speech technology still is considered an experimental input method, users don't have any real opinions about it, apart from an arbitrary ideological stance of whether the technology seems good or bad. Because of this, it was believed that the design process should be grounded in previous work to expand on successful design solutions as well as take lessons from the less successful designs.

A user test where users take active part in testing a proposed design hands-on, was chosen for its qualitative nature. An active user test encourages users to speak their thoughts which reveals a lot of insight into how they perceive the design proposal. While this generates a lot of qualitative data that needs to be interpreted, it is believed to be the best suit, because the outcome of such an evaluation, which will eventually assist in answering the research questions, provides the insight needed to understand what made the design successful or unsuccessful. Formal questionnaires or defining formal use cases with an accompanying test plan would work to evaluate the delivered functionality of the prototype which is irrelevant for this investigation since the research questions are related to users' perceived experience of interacting with a speech interface.

2.1 Literature review

A qualitative literature review for providing context framing as well as insight into the art of designing for speech interaction is conducted. Following the view of qualitative research described by (Cooper, Reinman and Cronin 2007), the outcome of the literature review is an increased understanding and deeper knowledge of previous related work. This understanding and knowledge is important to possess in order to efficiently and accurately investigate the research area. Moreover a formal literature review is not found to be suitable for this study because the purpose is not to identify, or create, perfect designs, but rather it is to create design proposals of fundamental interaction components for speech interaction that can be evaluated and turned into valuable insight for many projects.

“The goal is to gain general knowledge about the project's subject area (and perhaps related areas), and also deep knowledge about the particular problem that is being addressed.”

(Saffer 2007, 29)

The research in this thesis pertains primarily to the field of human-computer interaction but it is also influenced by the field of human-human interaction. For that reason, the literature review begins with a background analysis of these two research fields to provide a foundation of knowledge for the rest of the thesis' investigation.

The field of HCI is reviewed to identify the current state of research focusing on the design of user interfaces. The HHI review aims to form a fundamental understanding of how humans communicate verbally to each other by looking at the mechanics of a spoken conversation. The review of previous academic research is done to reveal insight into where current and previous efforts were focused as well as insight into the kinds of scenarios where speech interaction has been found to be the most appropriate. Additionally previous academic research is a source for learning about the reasoning behind interaction mechanics which adds to the fundamental understanding of what constitutes a successful speech interaction mechanics.

The review of commercial projects serves a similar purpose as the review of previous academic research in the sense that they both examine previous work related to speech interaction. Furthermore, their motivations are also the same; namely to better understand how successful speech interaction mechanics should be designed and in which usage scenarios speech interaction makes the most sense and is found to be the most useful. In addition to these shared goals, the review of commercial research also provides an aspect of the current state of the market which helps to form an understanding of how a speech interface should be designed to be found intuitive and graspable to use by end users, considering their current level of exposure to speech interaction.

In summary, the literature review's purpose is to gather the knowledge required to be able to create, through sketching, scenarios and prototyping, a sound design proposal for a speech interface that can bridge the design gaps defined in the research questions, RQ1 and RQ2.

2.2 Sketching

An integral component of any design process revolves around outlining features and characteristics of a proposed design. In interaction and experience design, this activity is referred to as sketching and can provide a lot of valuable insight into a design process. Sketches have been identified to be *"...helpful as visualizations of concepts and ideas that are still being formed to help clarify and communicate those ideas and concepts."* (Saffer 2007, 102).

The act of sketching can be done in several ways, however they all share the common purpose of illustrating the feel of a finished product. Instead of a literal definition of what a sketch is, a list of attributes aggregated by Bill Buxton (*Sketching User Experiences: Getting the design right and the right design* 2007, 136) roughly defines a sketch as something that is quick, suggestive and inexpensive. Considering these attributes and the overall reasoning present in (Buxton, *Sketching User Experiences: Getting the design right and the right design* 2007), the act of sketching is about concretizing ideas to make them easier to analyze and refine. The purpose is to fuel an iterative creative process where numerous ideas and concepts are included and evaluated to be able to eventually reach a thoughtful and possibly successful design concept.

In this thesis, sketching will be the most prominent activity of the conceptualization stage of the design process that will generate and refine concept designs for an engaging speech interface.

In order to be able to create a design proposal and prototype that can be evaluated successfully, sketching will be used to identify interesting scenarios and user tasks that people can relate to.

2.3 Scenarios

Scenarios are well-established and powerful tools that can be used to provide context for otherwise vague design objectives or strict product requirements. By utilizing a scenario it is possible to “*support reasoning about situations of use, even before those situations are actually created*” (Carrol 1999). This makes the design process more tangible and thereby more creative and fruitful since design decisions can be made based on how the design, in this case speech interface, is envisioned to be used, rather than an unpronounced or formal product vision.

Sharp, Rogers and Preece (2007) state that scenarios are also useful when performing user tests as they can be used to introduce test participants into fictive situations where the design makes sense, rather than just trying out a feature without understanding why they’d ever want to use a function like that. Furthermore, Sharp, Rogers and Preece provide a case study of a previous research project (Karat 1995) done on the topic of speech recognition, where each design feature was evaluated using feature-specific scenarios.

The investigation done in this thesis will make use of scenarios both to enable a creative conceptualization environment and to create a setting for user testing where the usefulness and value of speech interaction can be communicated and evaluated.

2.4 Prototyping

Building on the findings of the literature review stage and the concept sketching sessions, a prototype will be created to evaluate the concepts and ideas. A Hi-Fi prototype will be created to implement the outcome of the concept sketches into a testable artifact. As a Hi-Fidelity prototype, it will provide the same user experience as a potential future final version, but will not include fully functional application logic, only the interface components will be fully developed and implemented. Since the prototype will include a mix of off-the-shelf parts and new inventions, the Hi-Fi prototype will reflectively also be a mix of an off-the-shelf and do-it-yourself prototype as defined by Dan Saffer (2009).

2.5 User testing

User testing plays a central role in user centered design, and is an invaluable tool for finding out if a product performs the way it was originally intended to. By letting ordinary people, people outside of the design and development team, test the product, it is possible to extract plenty of information about every single part of a design concept.

The design process undertaken in this thesis project consisted of an initial conceptualization stage and a final evaluation stage. The evaluation was done through a user test that was made up of a set of user tasks which were specifically chosen to evaluate the design concepts which had been defined earlier in the conceptualization stage of the design process.

Sample size is a well-discussed topic among designers and engineers, some argue that quality comes from quantity while others claim the opposite. Soren Lauesen has taken a close look at this particular matter in his book (2005), where he conducted a usability test with ten users and compared the accumulated results from the first two user tests with the accumulated results from the last eight tests. According to his findings, almost all problems with a high hit-rate, problems that are reoccurring, would be discovered in a single user test. Therefore, he argues that for an initial test, it would suffice to only do one user test. However, Lauesen also points out that in the case of a single user test, it wouldn't be possible to distinguish if a problem is of high or low hit rate.

“After the fifth user, you are wasting your time by observing the same findings repeatedly but not learning much new.”

(Nielsen 2000)

Backed by his conference paper (Nielsen and Landauer 1993), Jakob Nielsen has published a web article (2000) where he proposes and explains why he believes that a maximum of 5 test users is enough. At the center of his research is a mathematical formula which he describes as follows: “...the number of usability problems found in a usability test with n users is:

$$N (1 - (1 - L)^n)$$

where N is the total number of usability problems in the design and L is the proportion of usability problems discovered while testing a single user. The typical value of L is 31%, averaged across a large number of projects we studied.” (ibid).

The formula shows that the greatest amount of insight is provided by the first three users and that the accumulated insight vastly decreases as the number of test users increases. Nielsen explains this further by stating that the results of the individual test users gradually overlaps with each other. The first and second test users' results has some overlap but it undoubtedly provides additional insight considering that all people are different. Results from the third test user considerably overlaps with the previous results and consequently contributes with only limited additional insight. Following this pattern, each test with a new user will contribute with less insight than the previous, meaning that the provided value decreases with the amount of test users.

In a more recent article (2012), Nielsen further backs his claim that five test users are enough. Based on statistics collected from usability consulting projects done at Nielsen Norman Group - a company that he co-founded with Donald Norman - he concludes that "...testing more users didn't result in appreciably more insights." (ibid).

The focus of the user testing conducted in this project is set to the perceived usability of the fundamental interaction mechanics, proposed in the concept designs. The user testing followed Lausesen's and Nielsen's parole that less is more, and targets an inclusion of five participants. Following (Cooper, Reinman and Cronin 2007)'s approach of qualitative research, the outcome of the user test is qualitative insight into how the concept designs supported their use of a computing device as well as understand how the proposed speech interface made them feel. This insight is vital in order to determine the usability of the concept design and ultimately to reach a conclusion for the thesis' overall investigation. While additional test participants would've yielded a better quantitative result, such as usability statistics, it was deemed inappropriate since the necessary insight would be attained faster and more efficiently using a qualitative approach of the user testing. Additionally, if a quantifiable number of tests were conducted and usability statistics were collected, their accuracy would likely not be as high, considering that the user tests are supposed to be carried out in the field, where the proposed concept are intended to be used.

In situations similar to this, one issue might be that while the location requirement may be formally defined as in the participant's own home, it cannot be assumed that everyone's home is the same. For instance, one participant might live alone, while another has a spouse and small children. Similarly, the test person might receive unexpected visits. These kinds of uncontrolled circumstances are common and can have unforeseen impact on the test outcome. A quantitative result may show this as a deviation or false-fact while a qualitative result may provide a more accurate result by describing the unforeseen and how it affected the user test. These results may even help to discover previously unconsidered aspects of a concept.

A test plan was created as a template and to provide formal structure to the user test (See Appendix: A). The test plan itself was based on the works of Jeffrey Rubin and Dana Chisnell (2008).

Overall test layout was the following: users were to receive general information about the project, an introduction to a scenario for setting the context and instructions regarding the user tasks which they're supposed to perform. The user test included a two-part semi-structured interview that took place before and after the participants performed the user tasks. The act of performing the user tasks represented the main focus of the user test. During this stage, the test participants were encouraged to think aloud and were left to explore the interface freely, following only the short instructions given for each user task.

3. Literature review

The literature review was conducted using Microsoft Academic Search (Microsoft Corporation 2014) tool which is a search aggregation tool that connects and catalogs most major journals' and universities' published content.

Searches were done using the following keywords:

HCI, HHI, NUI, GUI, UI, user interface design, speech, speech input, speech interaction, speech recognition, speech transcription, speech dictating, MMHCI, multimodal interaction, context specific computing, social computing, ubiquitous computing, pervasive computing, calm technology, feedback, sonic feedback, faceless interaction.

The outcome generated hundreds of papers originating from a multitude of research areas. An inclusion and exclusion process thinned out the collection of research papers to a more appropriate size of less than a hundred papers.

The criteria for inclusion of a research paper was that the paper had to investigate either design guidelines for speech interfaces or speech technology's applicability in real world scenarios. Furthermore, papers regarding related fields such as HHI or generic HCI were included based on their ability to provide fundamental understanding of their relevant fields. Papers not adhering to the criteria defined above, were immediately excluded from the review because of irrelevance to the project. Similarly, papers with a distinct technical focus were excluded since they are of little value to the thesis' investigation which' focus is to evaluate usability through interaction design rather than evaluate technical achievements.

3.1 Background

Using the sense of hearing as an input method can often be seen in science-fiction movies but have yet to become the norm for HCI. As humans' primary means of communicating messages is verbally, it can be argued that speech recognition ought to be a more efficient interaction method than keyboards and mice.

3.1.1 Traditional GUI design

Donald Norman, a pioneer in interaction design has conducted numerous studies of human behavior in relation to HCI. In his book (Norman 1998) he presents the “seven stages of action” which is an approximate model, explaining how actions are performed. The model says that actions are taken and intents are formed to reach a goal. Once an action has been taken, the consequences will be observed and evaluated to determine what has happened and what the next action should be. Norman presents the model in a “dotted list” format to emphasize that the seven stages does not necessarily have to occur in sequence, rather in any order, depending on the state of mind and perception of the surrounding world.

-
- *Forming the goal*
 - *Forming the intention*
 - *Specifying an action*
 - *Executing the action*
 - *Perceiving the state of the world*
 - *Interpreting the state of the world*
 - *Evaluating the outcome*

(Norman 1998)

A benefit from having knowledge about when, how and why actions are performed, is that designers get the second-best thing to looking inside the minds of users. By studying this model, it becomes apparent that in order for a user to find a system easy to use and intuitive, the user must be confident about what he or she is doing and what is currently going on in the system. This translates directly to the point of *forming the intention* and *perceiving the state of the world*. Users work toward a personal goal which may or may not be the same as the system's goal. If a user recognizes a problem that prevents him or her from reaching their goal, they form an intent to solve that problem. Thus, if a button needs pressing, then it should be made obvious for the user why it needs pressing.

Research such as this is important because it provides designers with insight into human behavior. From this insight, designers and researchers have been able to form guidelines for the design of HCI systems. From Norman's model above, it could be identified that: *it's important to inform a user how an action is helpful for reaching that user's goals, in order to make the interaction experience feel logical and intuitive.*

A widely recognized source of HCI design guidelines is Apple's human design guidelines, originally released and distributed in the form of a paperback book (Apple 1987). In that book, Apple describe how the Macintosh operating system uses a desktop metaphor to make it easier for humans to understand how to interact with and use a computer.

They provide application developers and designers with guidelines to follow when crafting application interfaces. The guidelines describe how interaction affordances and feedback should be presented to support the user. They provide guidance to visual design regarding things such as how to structure information and buttons, not only inside a window, but also how a window should look and behave. Icon appearances, menu-layouts and essentially everything else that is visible on the screen. This material was written to help third-party developers make applications that looked and behaved consistently with first-party Macintosh applications.

Apple's guidelines were published relatively soon after the initial conception of the WIMP paradigm and the desktop metaphor (Mott, The Desktop (Office) Metaphor 2003), and thus they were able to serve as a proof-of-concept design that following researchers and designers could relate to.

Although the knowledge gained in GUI design may still be useful, it is important to understand that it cannot be directly applied to novel interaction methods such as speech recognition (Bellotti, et al. 2002) (Nudelman 2013); traditional GUI-driven and speech-driven interactions are too fundamentally different.

Traditional GUI interaction makes use of human vision and cognitive abilities to present interaction affordances to its users. A speech interface doesn't necessarily require any visual components as the input method itself only needs a sound input. This means that GUI's depend on vision while speech UI's depend on sound. Because of this change of medium and human sense, the stage and possibilities for interaction are very different, hence requiring different interaction paradigms.

3.1.2 Human-human interaction

Within social science lies the Human-Human Interaction (HHI) field which investigates and aims to understand how humans communicate with one another. The most prevalent method for HHI is a spoken conversation.

3.1.2.1 Verbal and non-verbal messages

Human-human conversations was investigated by (Morency, Modeling Human Communication Dynamics [Social Sciences] 2010) who tried to model the communication dynamics of humans. In his investigation he talks about how people make extensive use of both verbal and non-verbal messages when communicating face-to-face. In a typical human-to-human, face-to-face conversation, most information is conveyed as spoken words (verbal messages). While the message "body" may consist of words, other factors, such as facial expressions and gaze, (non-verbal messages) are important as they provide context and feedback to the verbal messages.

The following sections are included to illustrate just how important these non-verbal messages are for a fluent human-human interaction.

Walkie-talkies provide its users with a communication medium that has sound as its single input and output modality. Because walkie-talkies operates on half-duplex channels, only one user may speak at a time, if multiple users speak at the same only one will be broadcasted and the other users who were talking will not receive anything as his device is busy trying to broadcast.

These characteristics means that walkie-talkies hinders all forms of non-verbal messages to be exchanged. To compensate for the lack of feedback, construct words such as “copy” and “over” were invented. These words does a good job at replacing the missing non-verbal messages that the participants need in order to know who’s supposed to be talking and if they understand each other. However, because human children generally learn how to talk by communicating face-to-face with their parents, the usage of non-verbal messages is firmly imprinted in people’s minds and this could reasonably be the reason why people may find it challenging to learn how to properly talk in walkie-talkies.

A step closer to full face-to-face communication is provided by ordinary phones. In what is relevant to this context, the main difference between a phone and a walkie-talkie is that a phone operates on full-duplex channels, making it possible for two people to talk and listen simultaneously. This means that while one person is talking (sending verbal messages) he or she can also listen for non-verbal messages such as “hmms” and “aahs” to know if the other person is following along in the conversation.

3.1.3 Multimodality

Looking back to the time before GUI’s, the common way to interact with computers was through command line interfaces (CLI’s). A CLI is a text based interface where users are supposed to type in any commands they want the computer to execute. Considering that CLI’s were text-driven, keyboards, which had already been successful in typewriters, came to be the preferred input method for HCI during the era of CLI’s.

As display technology evolved, computers were able draw graphics and display pictures. This led to the creation of GUI’s which attempted to make human-computer interaction easier. Within the desktop paradigm (see Chapter 2.1), the efficient text input capabilities of the keyboard was extended to also feature a pointing device, the mouse. By adding a mouse, and forming the desktop paradigm, users were able to more tangibly do office work on their computers. Selecting and moving documents around their virtual desktops were almost as easy as doing it in the real world.

This very brief history lesson above aims to point out that although the mouse and the keyboard are commonly regarded as one input method, they are two distinct input methods. Both of these input methods, or *modalities*, are frequently combined inside *multimodal* interfaces because their capabilities have been found to complement each other while working in tandem. As an example, the text of this thesis was typed on a keyboard while the images were positioned using a mouse.

“Everything is best for something and worst for something else.”

(Buxton, Multi-Touch Systems that I Have Known and Loved 2007)

With the recent development of “smart” mobile devices, the number of options available for interaction has increased from typical keyboards and keypads into more natural input methods. A natural input method is designed and developed to listen and act on natural body movements. An example is a gyroscope sensor which is able to detect the orientation of a device - is it being held upwards in portrait mode or sideways in landscape mode?

This development has enabled new interaction paradigms and forced designers and researchers to rethink HCI and explore the added possibilities on devices with multiple natural user input methods. (Nudelman 2013)

Touch is another natural input method that adds a large degree of tangibility to graphical user interfaces when compared to traditional mouse and keyboard input. Instead of using a mouse to control a pointer which in turn can manipulate screen items, touchscreens adds the ability to directly manipulate these with a human finger. A functionality that Apple took advantage of when designing the iPhone. (Apple 2007)

Being able to use multiple input methods interchangeably was one of the main purposes behind the design of Windows 8. Because Windows 8 was intended to be used on a very wide variety of devices, ranging from handheld tablet PC's where touch is the only input method, to traditional desktop PC's that rely solely on mouse and keyboard input, Windows 8's user interface had to be designed to support a multiple input methods. This resulted in a multimodal user interface that was designed to provide the best user experience when used with touch input but should also work satisfactory on devices using “legacy” mouse and keyboard input. (Microsoft 2013)

3.1.4 Sonic feedback

Interaction feedback can be delivered in different ways. In a GUI, the most prominent feedback is typically delivered visually to the user. For instance, pressing a button commonly results in the button transforming its visual shape to reflect its activation. Vision is not the only communicative medium used to provide feedback for an interaction. Feedback could also be provided audibly as sounds.

Absar and Guastavino (2008) provides a comprehensive description of the different types of sonic feedback that exists. They have divided them into the categories “auditory icons” and “earcons”. An auditory icon includes sounds that are tightly linked to an action, event or function, similar to visual GUI icons. They are often designed to mimic the sound of the object or event that they are representing.

Earcons, on the other hand, are generally more abstract and in contrast to auditory icons, earcons do not rely on skeuomorphism or tight event linking to become apparent for users. These sounds may thereby be harder to learn since the user needs to connect these sounds with an experience before they are able to understand them. The strength of using earcons

lay in their ability to provide richer feedback that can illustrate advanced or complex relations between actions, events and objects.

Stephen Brewster (n.d.), lists a number of advantages that the inclusion of sonic feedback in a user interface can bring. He recognizes its abilities of working faceless, being attention grabbing and interoperable with GUI elements. Furthermore, he goes on to motivate the inclusion of sonic feedback as a potential solution to avoiding visual cognitive overload induced by all interactions and information being presented visually.

The outcome of his research was that the inclusion of audible feedback allowed users to spend less attention on interacting with the interface and thereby being able to give more attention to the work being done. This because the cognitive load had now been split between vision and sound, instead of relying solely on vision.

A related research area that has employed sonic feedback is the topic of “calm technology”. Calm technology refers to a computing philosophy that aims to restrict the added cognitive load that people today are commonly subjected to from a plethora of stationary, mobile and ubiquitous computing devices, all requiring a user’s full attention. In order for technology to become “calm” a move towards an interaction model where devices and services can silently exist in the periphery of a user’s attention, only to appear when actually needed or wanted, is advocated. (Mark and Brown 1996)

In the context of calm technology, Bakker, van den Hoven and Eggen (2012), came to the conclusion that sounds are suitable for delivering feedback that can be absorbed in users’ peripheral attention field without disrupting their current focus.

3.2 Related work

People have been fascinated by speech based computer interaction for a long time. Speech UI’s are featured in several Science Fiction movies, TV shows and books (Noessel and Shedroff 2012). One of the most prominent and widely known examples is the omnipresent ship’s computer in Star Trek. By saying a control phrase, a ship’s crew were able interact with it through natural voice. In this case, using a control phrase made sense as it did not only provide a credible HCI pattern for a future system, it was also helpful in making the audience know who the actors were talking to as the computer was addressed in the same way as any of the human crewmembers. Once the computer’s attention was grabbed, they could engage in conversations by asking it questions or simply give it commands by telling it what to do (Paramount Pictures 1987).

“The ship’s computer must be addressed with a control phrase to get its attention, namely by saying “Computer”.”

(Noessel and Shedroff 2012)

3.2.1 Previous research

As is described earlier in this chapter, the most common way to interact with computers and digital devices today are through GUI's. The inherent communicative medium of a GUI is vision. Interaction affordances and output data are represented as visual elements such as text and graphics.

Similar to how GUI's are inherently vision based, speech UI's are based on audio, more specifically vocal audio. This is a fundamental difference and in direct comparison, it is observable that a speech UI does not necessarily have to make use of any visual elements neither for presenting interaction affordances nor for representing data.

3.2.1.1 Accessibility

An application area that has been extensively explored is accessibility. Because speech interaction doesn't inherently assume a visual component, the input method has been proposed and researched for use with visually impaired people who have difficulties interacting with GUI's. In addition to require sight, GUI's are typically designed to be operated using mouse and keyboard which requires precision hand and arm movements. This can be difficult for people with upper-body motor impairments as they may not be able to manipulate the mouse and keyboard input devices correctly and are thereby unable to use a digital device successfully.

In (Mustaquim 2013), the increasing number of elderly people is identified as a motivation for investigating how digital devices, particularly games, can be designed to include a broader demographic by also targeting elderly people. Considering the pervasive role IT has come to play, Mustaquim expresses that it would simply be unsustainable to exclude a large demographic from the benefits provided by IT only because designers don't know how to design for them. In his project he proposes automatic speech recognition to be a viable input method because it enables an auditory communication channel which may be more appropriate for people with disabilities, such as the elderly.

Speech interaction's inherent characteristic of relying on audio and spoken words has appealed to a lot of designers and researchers as it imposes a possibility of crafting HCI experiences that are as fluent as speech based HHI.

Furthermore, using voice and existing human languages to interact with digital devices can enable people who don't know how to read and write to be able to take advantage of the benefits that are provided by digital technology. Using their native language they would be able to simply tell a digital device what to do. Depending on the application, output could also be delivered as spoken words in their native language. (Brewster, et al. 2011)

3.2.1.2 Transcribing speech - dictating instead of typing

Another aspect of speech recognition to consider is the ability to transcribe spoken words in real-time. Based on current GUI-based interface design patterns, (Kalnikaitė, Ehlen and Whittaker 2012) have created and evaluated a speech operated tool for making annotations

during meetings. With the intention of creating as less of an intrusive experience as possible, they wanted to develop a tool that could help ease the cognitive load of meeting participants by allowing them to create annotations without the considerable effort required when typing. Using automatic speech recognition, their prototype tool was able to transcribe everything that was said during a meeting. When running, the tool's GUI presented one clearly visible button that allowed the user to highlight a section as important. Post runtime, a meeting transcript complete with annotations is available.

The work of Kalnikaitė, Ehlen and Whittaker, demonstrates how different input modalities can work together to produce end user value. Using the audio channel for documenting meeting conversations and the vision and motor channel for controlling the documentation process.

Speech recognition has the ability to transcribe spoken words into text strings. Because of its potential of being able to vastly outperform the number of words per minute, WPM's, a user can generate on a traditional keyboard, text dictating has been touted as viable replacement or alternative to them (Yuan, Liberman and Cieri 2006).

This potential was explored by Hoste and Signer (2013), who identified an emerging design gap in products that were not originally intended for heavy text input. These products include smart TV's, public information displays and game consoles. Using speech recognition and in-air hand gestures, they propose models for efficient text entry without using physical keyboards. A user uses its voice to continuously dictate the text that is to be entered into the device. As the user speaks, he or she can wave his or her hands to highlight missinterpretations and have them quickly corrected to a word the system believe is similar, alternatively they can spell a word letter-by-letter if necessary.

Through mixing two different NUI input methods they have created a multimodal interface where the voice medium and hand movements are used simultaneous to enable a fluent input method for text. Speech provides the bulk of data, as in words, while hand gestures provide active feedback and correction.

From user tests, they have concluded that their speech and gesture enabled solution provides an input method capable of being at least 4 times faster, higher WPM scores, than comparable controller or gesture based keyboards. (Hoste and Signer 2013)

3.2.2 Previous commercial work

Commercial speech recognition products have been developed for a long time and consequently they have changed shapes a lot over time. Early work largely provided the ability to control a GUI using speech while more recent products are intended to be used more stand-alone with a specific and dedicated interface.

3.2.2.1 Dictating with “Dragon”

One of the earliest is *Dragon naturally speaking*. The Dragon software first appeared in 1975 when James K. Baker published a research paper describing a method for transcribing spoken words into strings of text (Baker 1975). Since then, his research materialized in a commercial product that aims to replace keyboards with speech recognition software. Given its roots in

the 1975 research article, Dragon's primary functionality has been to give people the ability to dictate any text they want entered instead of having to type it on a keyboard. Dragon's target user group is people who make extensive use of keyboards and could benefit from having another input method available to achieve a more ergonomic workplace. Later versions of the software also added the ability control a computer using voice commands. Today the Dragon product belongs to Nuance Communications Inc. who has expanded the product portfolio to include solutions for additional applications such as interactive answering machines and healthcare (Nuance 2014).

3.2.2.2 Personal assistant "Siri"

Powered by Nuance's speech technology is Apple's Siri software (Wildstrom 2011). Siri is a personal assistant-like application that is accessible through a long press on the home button of an iPhone or iPad. After the key press, a full screen GUI appears along with a textual greeting, asking you "What can I help you with?" (Apple 2013). The user responds by asking Siri a question or telling *her* what to do. Siri responds verbally with either an acknowledgement that an action has been taken or with an answer to the user's question. A user's recognized speech, user request, and Siri's responses are displayed in a conversational styled GUI where the users input is visible to the right and the system's output is visible to the left. Alongside the conversational GUI design, Siri is also able to understand different language semantics and is able to reason what is likely to be the user's intended question. It may for instance conclude from the current GPS location which restaurants you're likely to be looking for or make use of data from social media to know what number to dial if the user asks for a call to "his wife".

While the main focus of marketing for Siri revolves around its abilities as a personal assistant, like keeping track of a calendar, Siri also provides the ability to work as a launcher for applications and for transcribing text into messages. (Apple 2013).

3.2.2.3 Speech platform "SAPI"

Microsoft has for a long time provided developers and end users access to speech recognition technologies. The Speech Application Programming Interface (SAPI) provided developers with an abstracted and standardized platform upon which they could build their own speech enabled applications. The platform provides both speech recognition as well as text-to-speech functionality. (Microsoft 2009)

3.2.2.4 Dictating with the "Input panel" in Windows XP

Several speech applications have been developed on top of the SAPI platform including the input panel found in Windows XP Tablet PC Edition, a specialized version of Windows XP that was tailored towards mobile computers with touch and pen input support. The most notable difference compared to the conventional Windows XP editions was the aforementioned input panel. From this panel, users were able to enter text using a stylus, an onscreen keyboard, or

speech. The input panel resided in the system taskbar next to the start button and folded up horizontally above the taskbar where it provided NUI input to both first- and third-party applications. (Thurrot 2004).

3.2.2.5 Fully voice enabled GUI in Windows Vista

Based on the SAPI engine and the work done in Windows XP, Microsoft released a new incarnation of speech input called *Windows Speech Recognition* with Windows Vista. In this version, the voice capability of the input panel had been replaced by a standalone speech “agent” that was omnipresent and dockable to the top of the screen. The agent provided a GUI button for enabling and disabling speech input as well as visual feedback on the recognized command and microphone sound level. In addition to the GUI buttons, users could also use the voice commands, “Start listening” and “Stop listening”, for toggling speech input. For this agent, every aspect of the Windows GUI was reworked to support speech interaction, application switching was done by saying “Switch to Application X” and buttons were “clickable” by saying their labels. In addition to voice commands, Windows Speech Recognition also supports dictating for entering paragraphs of text. (Microsoft 2006)

3.2.2.6 Cloud based speech platform

The first version of the SAPI shipped with Windows 95 and the latest shipped with Windows 7. With Windows 8 the same version of SAPI and Windows Speech Recognition was still included but also accompanied by a cloud based service called Bing Speech. While SAPI provides just about the same functionality and experience as it did in Windows 7, the Bing Speech API provides developers the ability to incorporate speech into their own applications. From a user experience designer’s perspective, the main advantage of Bing Speech is that it doesn’t require a lot of local hardware resources as the actual computation is performed in the cloud. This means that speech recognition is able to run on devices with low performance as well as help preserve battery on mobile devices. Moreover, the SAPI platform is based on user profiles that get trained to enable more accurate speech recognition. Bing Speech does not have this requirement and works immediately for anyone. (Microsoft n.d.)

3.2.2.7 Voice controlled “launcher” in Windows Phone 7

A different implementation of speech interaction can be found in Windows Phone where there are no visible GUI cues to indicate voice recognition functionality. Instead, users are supposed to press and hold the physical Windows button on their device until a speech GUI with instructions appears on their screens. With this approach the speech UI is not so much for controlling individual applications as it is for controlling the phone itself. In this incarnation, one may consider the speech interface of Windows Phone as an application launcher that lets users do things like, making phone calls, texting friends, searching the Internet or launching applications. (Microsoft 2012)

3.3 Conclusion

As is demonstrated by popular science fiction works, such as Star Trek, speech recognition technology is a research area that has been and still is of interest to a wide range of people. Although science fiction is just that, fiction, both academic and commercial researchers have investigated the area extensively. Yet, speech recognition research and development has a long way left in terms of making it *useful* and thereby used by people.

Previous work has made huge strides in making the underlying speech recognition technology function accurately and reliably. Typically, the work is done by traditional computer scientists whose' main motivations and goals are mostly technical and quantifiable, such as reaching a certain level of accuracy under a specific condition.

There have also been research motivated by less technical goals. The most prominent non-technical motivation for researching speech recognition is to help people with disabilities more easily accomplish everyday tasks that otherwise might be hard or impossible for them to do because of a disability.

Speech recognition in its current state provides invaluable help to a lot of people all over the world. It serves as a bridge across the gap between humans with reduced visual or motor abilities and traditional computers that rely solely on vision and fine motor skill. Additionally, speech recognition is also able to connect people more easily with the information and services they need, and also with other people.

Because a typical graphical user interface relies on graphics and text to communicate with their end user, end users have to be able to read, this is often taken for granted in developed countries but is a major problem that effectively blocks people from using and benefitting from modern IT services.

In commercial research, a recent trend is to make speech interaction useful for a larger demographic, not only including people with disabilities, that is. Efforts from both Microsoft and Apple show that there is an industry-wide intention of bringing speech recognition to everyday devices, such as smartphones and PC's, as a complementary or alternative input method. The purposes are to enable more convenient and seamless HCI experiences in situations and contexts where traditional GUI's might be bulky, inefficient or even dangerous.

In conclusion, the research area is vast, ranging from reaching technical achievements to investigating social contexts. A lot of technical work has been done to make the technology function while there is a lack of understanding the when, where, how and whys of interacting using speech. Accessibility research and products have established that speech recognition technology is feasible and can be useful in real world situations.

Recent commercial projects, including Apple's Siri, have begun exploring use cases and situations where speech interaction may make sense. However, since these commercial products are not preceded by a lot of previous research in those areas, it is a young field with several aspects left to investigate.

This thesis' aim is to investigate at least one of these aspects, namely the "how", in terms of how should I give a computer a speech command and how should I receive the corresponding output.

4. Design process

Initial steps of the project involved exploring and defining suitable scenarios. Choosing an appropriate context was considered an utmost important activity as it was believed to have a profound effect on the perception of the delivered prototype and research investigation as a whole. Selecting an approachable scenario that a lot of people can relate to would not only make the report easier and more sensible to read, it may also make user testing more efficient as testers are less likely to be distracted by an unfathomable arbitrary purpose. The process of identifying and selecting scenarios consisted of a literature review and several sketches. The literature review was used to learn about previous research and existing commercial products. The sketches, mockups and technical proof of concept prototypes were used as tools for interpreting the gathered information. Additionally, the sketches and mockups were made in hi-fidelity which made them a valuable instrument for exploring the probable experience provided by different conceptual designs.

Technical proof-of-concept designs were created to evaluate different speech recognition engines and platforms. These designs were purely technical and thus they did not focus on user experience. Instead they played a key role in the development of the hi-fi prototype as they were used as technical reference designs.

The design process' final stages involved the design, development and user testing of a hi-fidelity prototype. This prototype was derived from the preceding stages of user experience sketching and technical proof-of-concept development. Essentially, the hi-fi prototype was created to bring the user experience sketches to life by making them interactive. Although complex, the development of the prototype was a straightforward effort where the interaction design as well as the technical implementation had already been decided in the previous stages of the design process. This meant that focus could be set on delivering a well-polished prototype that was able to carefully deliver the intended user experience. Up to this point the design process had not involved any user participation, literature review and design sketching had been used to identify and propose a solution for a design gap. With a working Hi-Fi prototype, it was possible to evaluate the design proposals using user tests. Outcome of the user tests would be used to make a conclusion and ultimately answer the research question.

4.1 Conceptualization

A motivator for investigating speech recognition as a method for HCI is its renowned potential of enabling easy and fluid computing experiences. Therefore it deemed important to work with a scenario that could demonstrate how speech input might make sense. In order to identify such a scenario, the project has undergone an extensive conceptualization process where previous work and related products have been examined to help better understand what might work and what might not. The other activity involved in the conceptualization has been the sketching of potential interactions and interface designs. Sketches have played a key role in exploring design concepts and scenarios. In contrast to textual system requirement documents, a sketch has the advantage of being more tangible and easier to understand and relate to.

4.1.1 First round of concept sketching

The user interface or shell of Windows 8 was designed to support multimodal interaction. It was optimized for touch while also being usable by mouse and keyboard. This design allows users to move interchangeably between any available input modality as they seem fit. Expanding on this ability, it was chosen to propose speech interaction as yet another input modality for the Windows 8 shell. Adhering to the modeless design of Windows 8 has endorsed the use of an omnipresent speech interface that would be integrated for pervasive access throughout the operating system. Furthermore, this design decision was based on an idea of creating a relationship between touch and voice input that is similar to the one between the mouse and the keyboard (see Chapter 3.3).

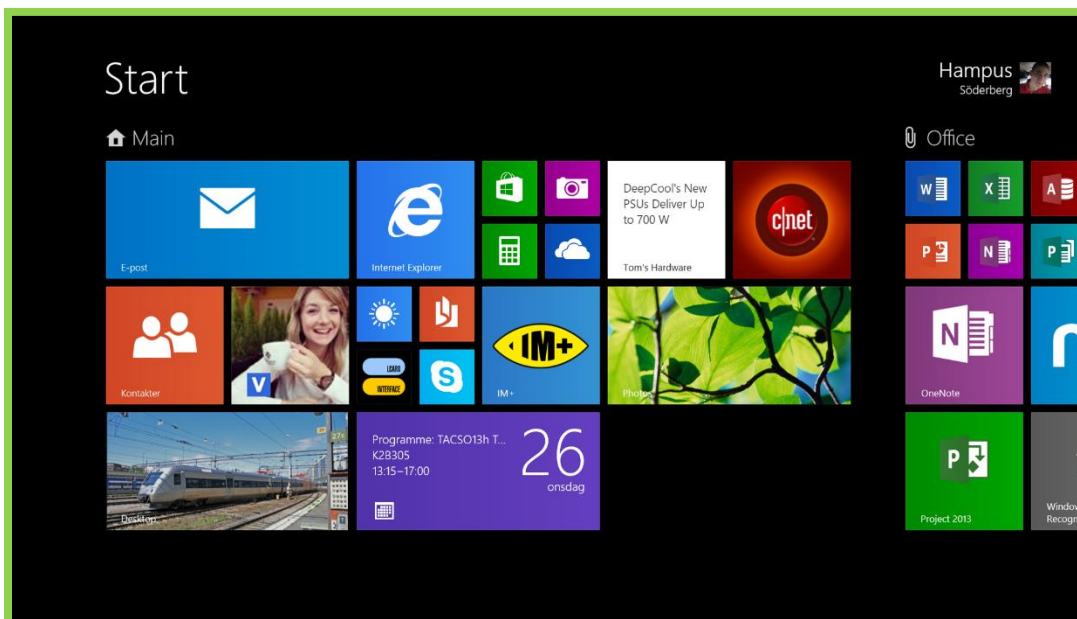


Illustration 1: A voice activated Windows 8 UI without visual hints.

These concepts relied on the notion that the user would be aware that speech input is universally available. Furthermore, it was also taken for granted that a user would intuitively know what to say. In essence, this design assumes that the underlying system is intelligent enough to be able to interpret arbitrary voice commands from an end-user.

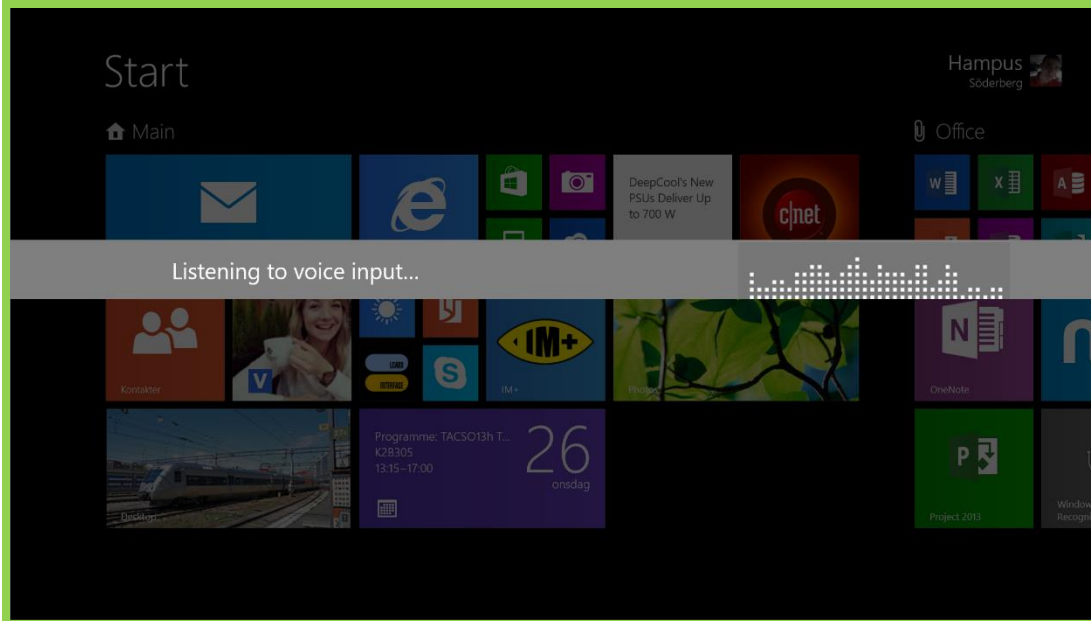


Illustration 2: Voice activated Windows 8 UI with the "feedback ribbon".

They did however include visual feedback elements that aimed to assure users that their commands were heard and accurately recognized. The first feedback sketch was based on the same design as Windows 8's default modal dialogs and was dubbed the "feedback ribbon". When reviewing the sketch it was found to be too intrusive and might cause users to lose track of their main activity.

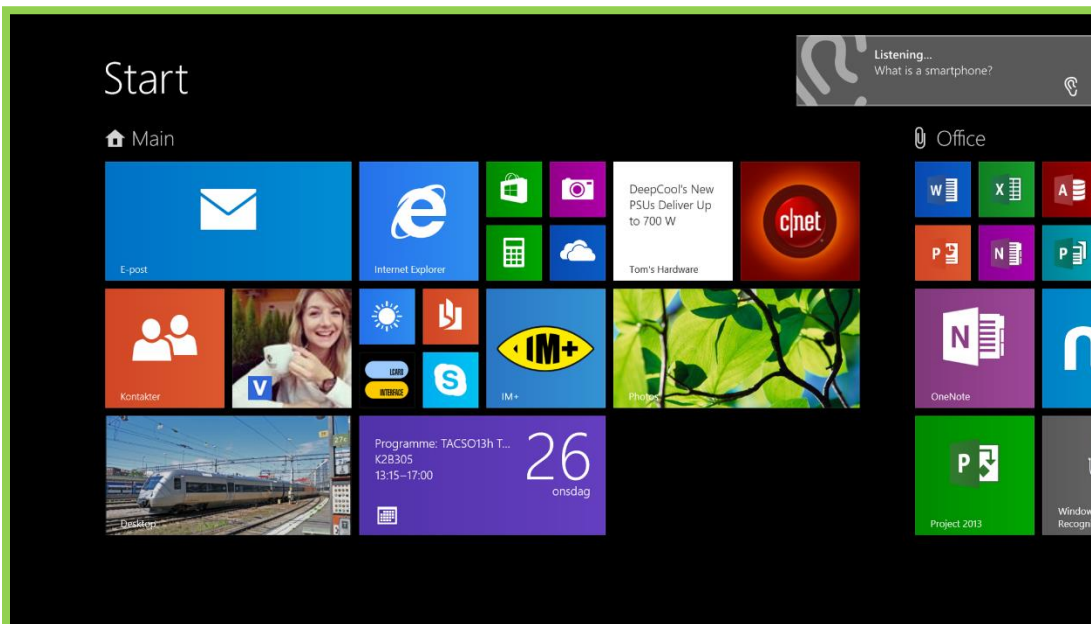


Illustration 3: Voice activated Windows 8 UI with "feedback toast".

In an effort to promote a more seamless modality intersection, the feedback ribbon was replaced with a more subtle "toast" in the top right corner. In Windows 8, a toast is an

interface element used for delivering notification style messages such as notifying the user when an email is received. In this concept, the toast design is borrowed to also provide functionality similar to the feedback ribbon. When the computer hears a user's voice it displays a toast to indicate that it is listening. Additionally it provides continuous feedback by printing out the user's spoken words as they are recognized by the computer.

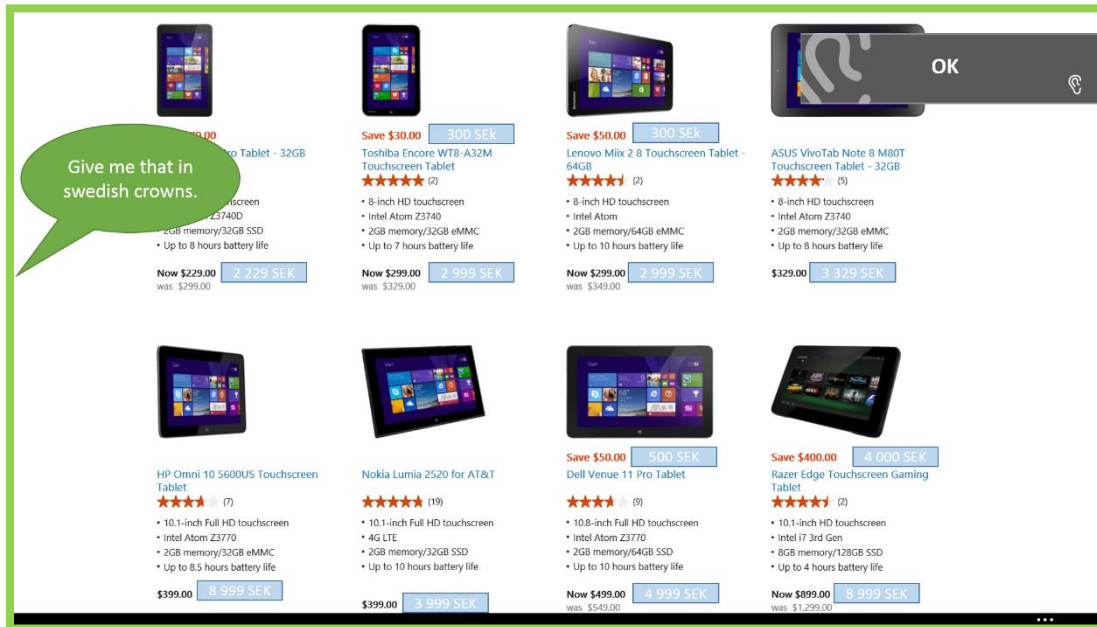


Illustration 4: Voice activated Windows 8 UI with "dynamic action".

This concept was referred to as "dynamic action". As the name implies, this function was intended to demonstrate how an omnipresent voice interface could seamlessly enhance a common touch or mouse and keyboard experience. By utilizing speech, a user doesn't have to leave context searching for a currency converter service, instead a voice command can execute this task in the background and integrate the output data as a visual overlay on the contents at front. Furthermore, a request like this might generate a verbal response and effectively avoid any additional cognitive load from interface visuals.

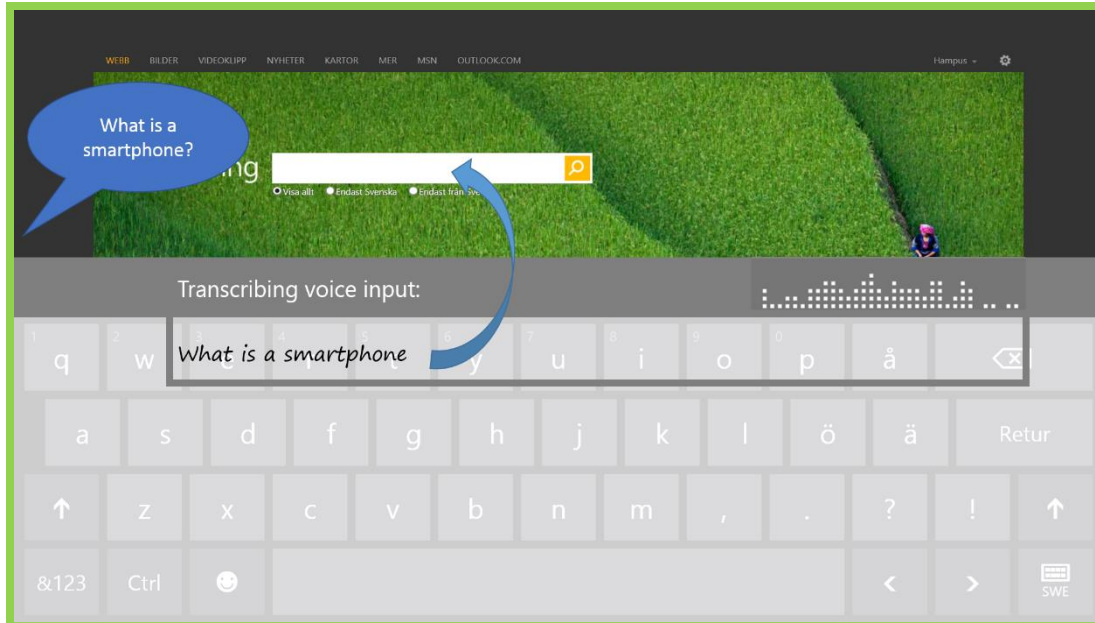


Illustration 5: Voice activated Windows 8 UI with "voice keyboard".

Transcribing voice to text was identified in the previous research and previous commercial work sections of the literature review to be a recurring topic of interest. This functionality has the potential to outperform traditional keyboards, both virtual and physical alike. This made it a sensible choice to include in the Hi-Fi prototype.

The feature is thought to be tightly integrated with the system so that whenever a textfield gets focus, any speech input is directly transcribed into submittable text. As the illustration 4 shows, using dictating should not replace keyboard input but rather extend it by integrating a speech dictation functionality.

4.1.2 Second round of concept sketching

The first round of sketching resulted in several mockup designs of potential solutions and scenarios. These designs went through an iterative process whose aim was to generate a design solution for the final Hi-Fi prototype. Since the purpose for making the Hi-Fi prototype was to evaluate the perceived user experience, the Hi-Fi prototype was designed specifically to be suitable for user testing. Subsequently, because the Hi-Fi prototype's design was derived on the outcome of this second round of sketching, the aim for this round was to identify scenarios suitable for user testing and to specify the possible design solutions to evaluate.



Illustration 6: Voice activated Windows 8 UI using a toast to present a speech input modality.

Among the ribbon and toast designs the toast with its smaller cognitive footprint appeared to be the best choice for exposing speech input functionality. In previous concepts, there were no clues of any kind that could indicate that the computer's shell supported speech input. This was revised in this sketch iteration as it was believed to be hard or impossible for a user to figure out that they could interact using their voice. This belief was grounded on the current state of the market where speech interfaces are rarely used, and if they are, they are typically provided as a go-to feature that users manually launch when they want to. The speech interface proposed in this thesis works in a different manner. It is omnipresently available from startup to shutdown and thereby doesn't require the user to take direct action to enter a speech input mode. To expose the speech affordance, the toast design was adopted to provide users with a notification-style message instructing them that their computer listens for speech input. In the process of re-purposing the "feedback toast" towards an affordance toast, the interaction feedback mechanic was changed to no longer rely on graphical elements but instead rely solely on sounds. A pitch-up sound played when the affordance keyword, i.e. the computer name, was heard and a pitch-down sound played when the computer stopped listening for speech input. This change of direction was motivated by the conception that a speech interface is audial and thus should not need graphical components for its interaction mechanics. To engage the speech interface, the user should address it in the same manner that he or she would address a human. The default state of the speech interface is to not listen for any command until it's been directly addressed. Consequently, to get the computers' attention, the user should say its name. Once it has heard its name it will begin to listen for a command. To help screen out misrecognitions, the interface will only listen for roughly half a minute or until a command is recognized. To regain its attention, the user would have to address the interface with the computer name again. A successful voice command would consist of the computer name, followed by the action to be performed and details specifying the action. An example would be "HAL, open web

browser”. More on this in Chapter 5.2.

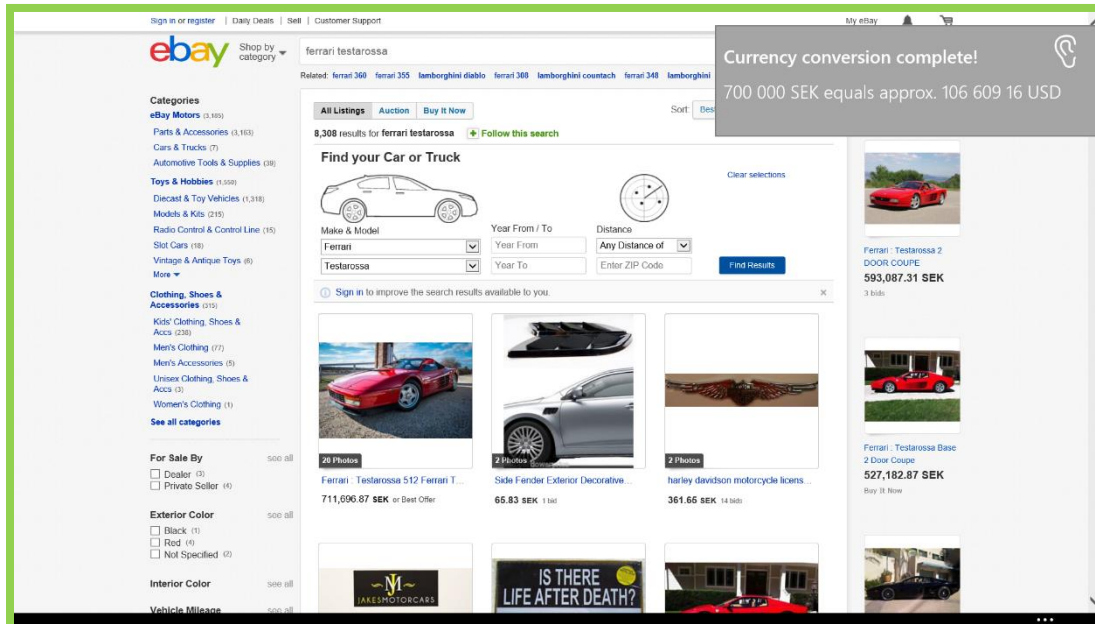


Illustration 7: Revised version of the “dynamic action”-concept.

The decision to use the toast design also influenced the “dynamic action” concept. Due to technical delimitations of the thesis’ scope, to achieve better consistency throughout the prototype and to better integrate with the host operating system, the ability to automatically extract and overlay information on a web page was replaced with a “quick question, quick response”-feature.

This new Q&A concept revolves around the user being able ask the computer a seemingly simple question and receive a concise answer to that question. Seemingly does in this case refer to something that would include a single task and result in a single answer. The concept identified asking for a currency conversion of a price as one such question. Upon asking, the user receives a toast displaying the converted price.

The intention of including the original dynamic action concept was to demonstrate how speech input might provide context-neutral shortcuts. Although technically less complex, the quick answer, quick response concept provides the same kind of shortcut functionality while preserving interface design coherency.

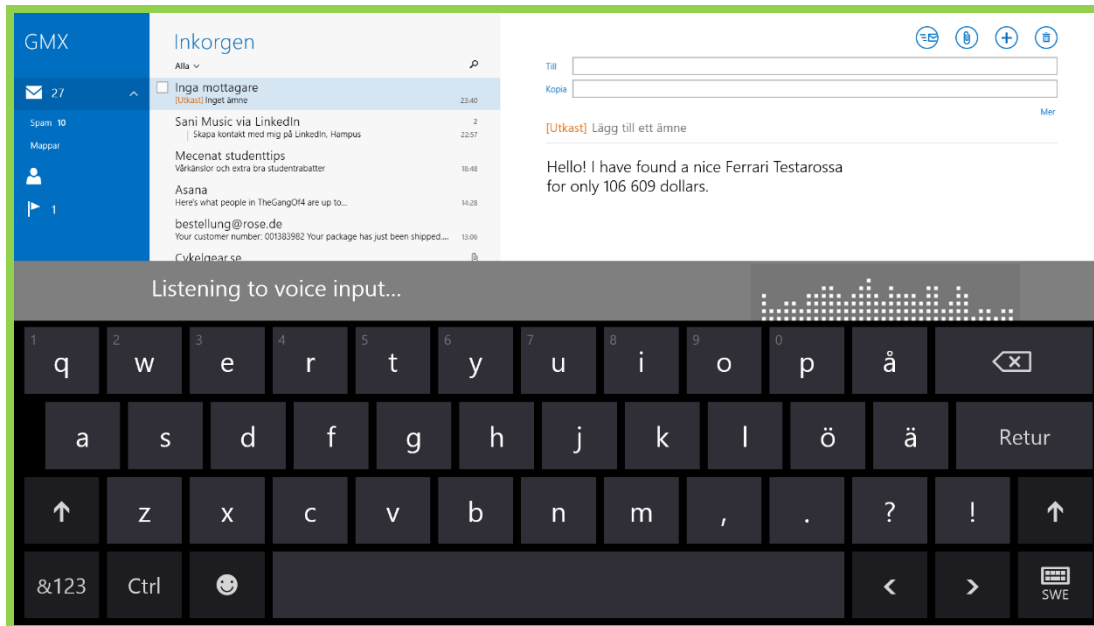


Illustration 8: Revised version of the “voice keyboard”-concept.

The “voice keyboard” concept retained its initial design that was to have it integrate with the onscreen touch keyboard. The conceptual voice keyboard could be used anywhere text input is required but for illustrational and demonstrational purposes it received a use context in a scenario. The scenario was to “write” an email message using voice.

4.2 Technical Proof-of-Concept

Two technical proof of concept applications were developed to ensure the project's technical feasibility. The applications were created with the sole purpose of exploring technical platform capabilities, and did subsequently not involve any experience design. Instead the development of these applications focused on software design and aimed to create reference implementations which could later be used for building the Hi-Fi prototype.

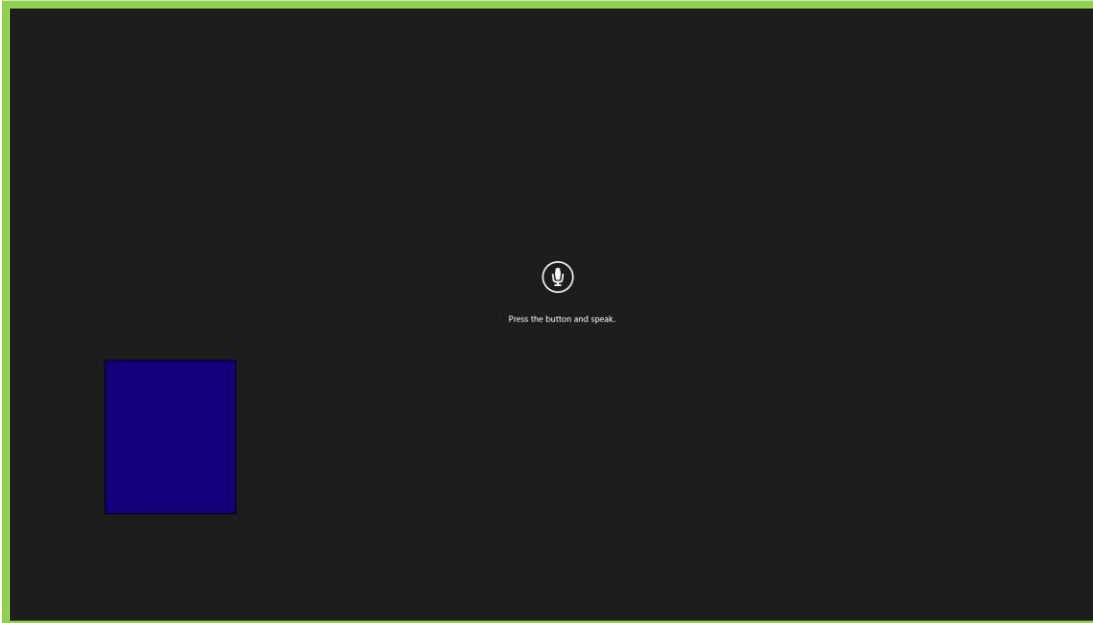


Illustration 9: Technical Proof of Concept using Microsoft Bing Speech.

The first proof-of-concept application features a button that effectively turns the speech recognition functionality on or off. When active, the application listens for a command to move the rectangle to the right. Once the application recognizes the correct voice command it will move the rectangle. This proof-of-concept implements Microsoft's Bing Speech SDK which provides the underlying speech recognition technology. In this implementation, default SDK components for engaging a speech interaction and for receiving relevant feedback are used.

The Bing Speech platform provides speech-to-text as an on-demand service. It works by opening an audio stream to an online service that in turn sends back the transcribed speech. Although the technology is outside of the scope for this thesis, it is interesting to note that it is powered by a cloud service, Windows Azure, and is thereby able to run on low-performing and energy efficient devices. This means that from a purely technical standpoint, the findings of this investigation is implementable on everything from desktop computers to embedded devices.

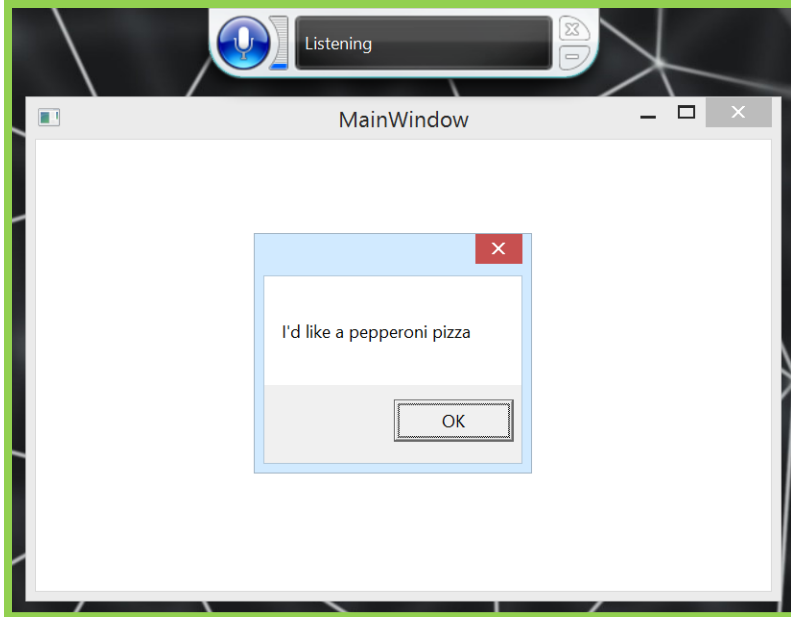


Illustration 10: Technical Proof-of-Concept using Microsoft SAPI.

The second proof-of-concept application made use of the speech recognition engine SAPI that is built and used by Microsoft for Windows Speech Recognition (see Chapter 3.4.1). In contrast to Bing Speech, Microsoft SAPI does not rely on a remote server backend for transcribing speech. Instead it provides a full speech recognition engine that runs on the local computer. While requiring more local resources, it has the advantage of being able to continuously listen for an indefinite length of time. Bing Speech does not provide continuous listening as it follows a transaction-like model where every speech transcription action is treated as a request. Additionally, by design these requests must be triggered by something other than a voice command.

This application demonstrates a working implementation of a SAPI speech recognition engine. Without going into specifics, the SAPI engine uses a concept of dictionaries to handle what words and phrases the interface will listen for. The above illustration shows a dictionary containing pizza phrases in action. Once the recognizer engine hears a word or phrase that exists in the dictionary, the application will display a message box with the recognized word or phrase.

4.3 Hi-Fi prototype components and mechanics

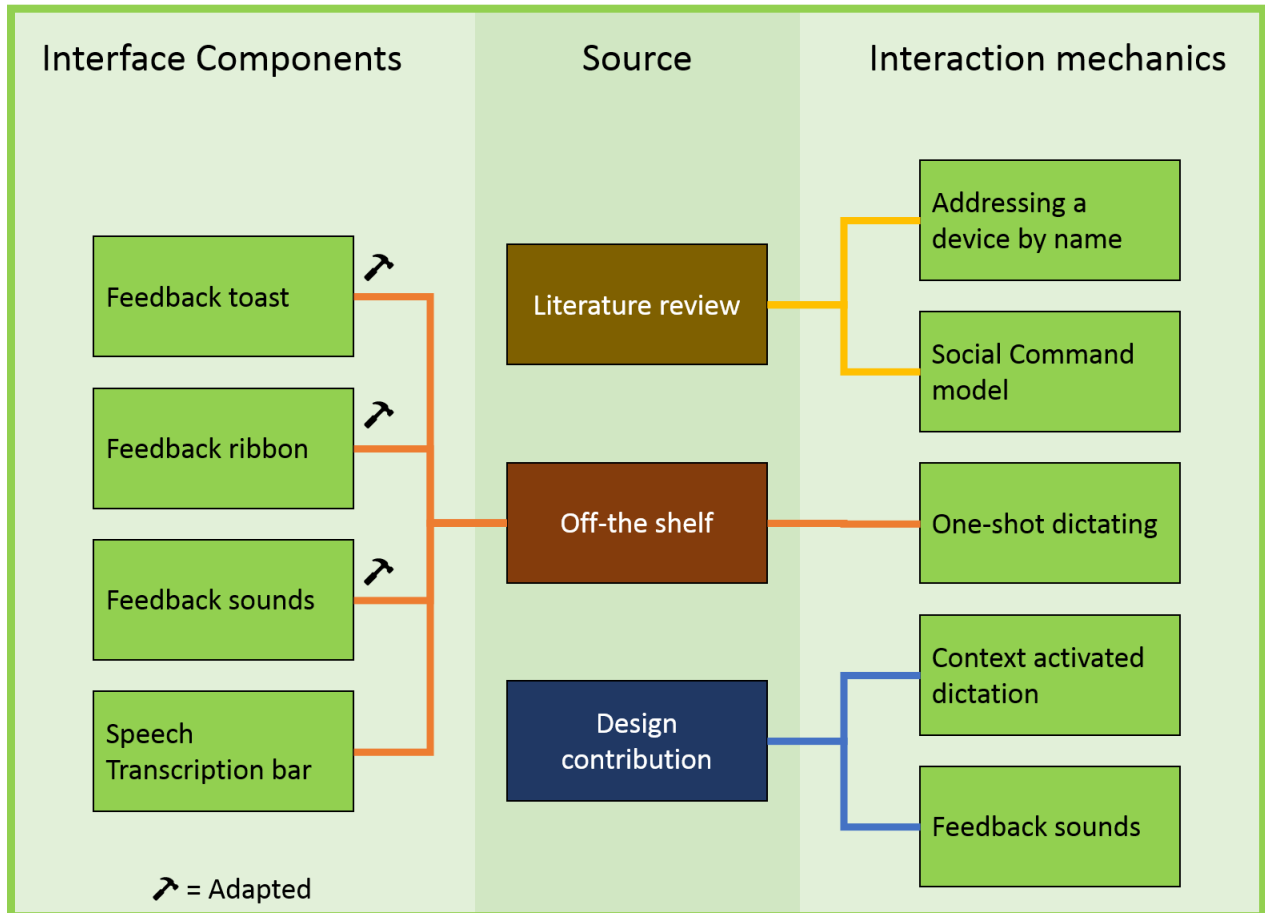


Illustration 11: Speech interface components map.

4.3.1 Interface components

The Hi-Fi prototype was built using a mixture of existing, adapted and new components. As is depicted in Illustration 15 above, all included GUI elements were either implemented directly as off-the-shelf parts or modified to serve a purpose, different than the one it was originally designed for.

4.3.1.1 Feedback ribbon

The feedback ribbon was based on a default Windows 8 component which was originally used as a modal dialog for prompting users to take action, make a decision or explicitly acknowledge an important notification message.

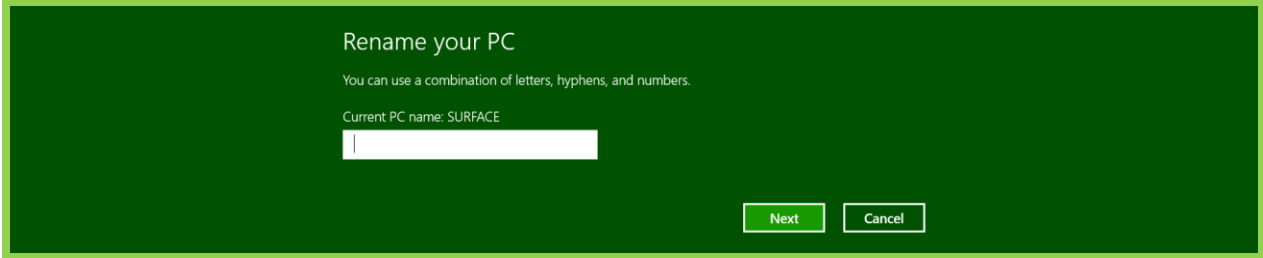


Illustration 12: Default Windows 8 modal ribbon dialog.

Its visual design was roughly the same while its behavior changed rather substantially. Instead of prompting a message that needs responding to, the ribbon would appear whenever the speech interface was listening to provide continuous recognition and interaction feedback. The ribbon would automatically disappear after a successful or cancelled speech command.



Illustration 13: Conceptual feedback ribbon.

4.3.1.2 Feedback toast

Similar to the feedback ribbon, the feedback toast was a re-purposed Windows 8 component originally called a “notification toast”. While its behavior and visual appearance remained the same, its use was changed. The intended use for the notification toast is to show short informative messages or alerts to users. An example would be to notify a user of a new e-mail message.

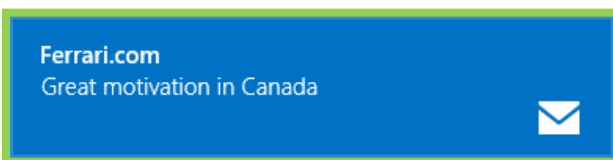


Illustration 14: Default “notification toast” used to notify user of a new e-mail message on Windows 8.

In the Hi-Fi prototype, the notification toast was dubbed the “feedback toast” and was originally intended to be used, displayed, whenever the computer was actively listening. Throughout the design process its use was changed and it was later used to notify users of the speech input affordance as well as to provide visual information output.



Illustration 15: Modified notification toasts which act as affordance presentation and information output elements.

4.3.1.3 Feedback sounds

The feedback sounds implemented in the “launch and getting answers” portions of the prototype were un-modified default Windows components. They are used by default in Windows speech recognition to indicate a similar - but not identical - transition between a listening and a not-listening interaction mode.

Furthermore, the feedback sounds used in the “speech for writing” portions of the prototype were also off-the-shelf parts which came as components from the Bing speech SDK.

4.3.1.4 Speech transcription bar

Bing Speech provided a fully-featured speech recognition GUI, referred to as the “speech transcription bar”. The bar would appear as an ordinary app-bar on the bottom of the screen whenever a speech interaction had begun, to provide continuous visual feedback to the user.



Illustration 16: Default Bing speech transcription GUI.

The transcription bar itself was left almost entirely unchanged except for its activation (see Chapter 5.4.3.4) and positioning. Instead of being presented as part of an individual application’s UI, its position was coupled to the onscreen touch keyboard to make it appear omnipresent and integrated with the system’s UI.

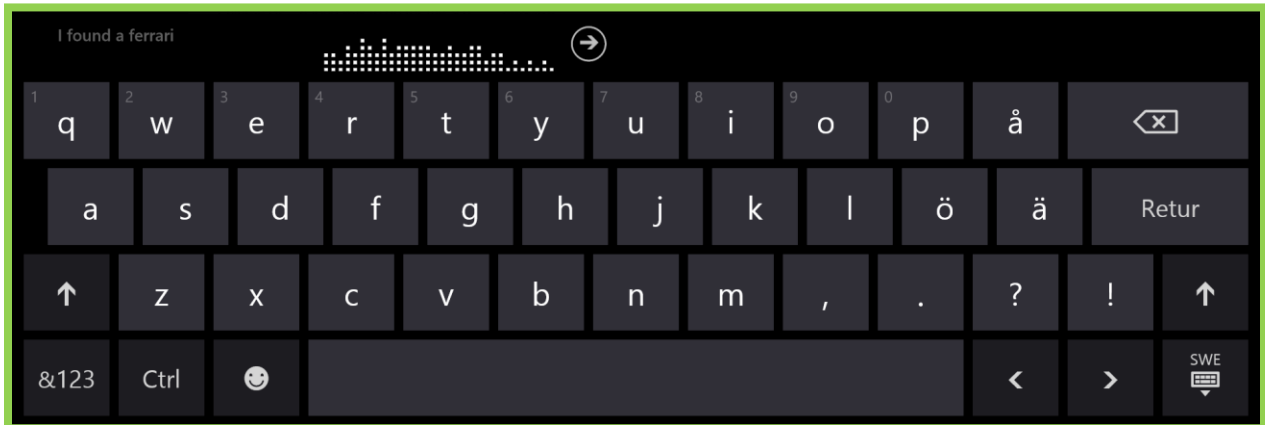


Illustration 17: Default Bing Speech transcription GUI “docked” to the default on-screen touch keyboard.

4.3.2 Interaction mechanics

The Hi-Fi prototype was created to evaluate how useful and intuitive the proposed speech interface was perceived to be. Therefore, several interaction mechanics were carefully implemented in the prototype. Compared to the GUI components, the interaction mechanics were more varied and were conceived based on a variety of sources. Some mechanics were

based on previous or current commercial products while others were derived from research findings. Other interaction mechanics were new design contributions conceived during the sketching stage and as such were loosely grounded in the literature review.

4.3.2.1 Addressing a device by name

The literature review found an example in popular culture where a computer should be addressed by foregoing any command with a “computer” keyword. This mechanic in itself was kept intact to investigate its feasibility in real world applications.

The mechanic was however modified to move away from its original formal tone towards a more humble one. This was achieved by modifying the attention keyword. Instead of saying “computer”, the user should say the name of the particular computer or other digital device he or she wants to interact verbally with.

The intent was to adhere to the mechanics of everyday HHI communication where people commonly say a person’s name, rather than “human”, to get his or her attention and ultimately be able to engage in a conversation.

The Hi-Fi prototype used this mechanic whenever the “speech for launching and getting answers” parts of it were used.

4.3.2.2 Social command model

The “subject -> predicate -> object” social command model worked in symbiosis with the “addressing a device by name” mechanic described above. Similarly, this mechanic could also be found in the popular culture reference, but in contrast to the “addressing a device by name” mechanic, the social command model was not created to evaluate a fictional interface mechanic. Instead, the social command model was conceived during concept sketching based on the notion that using a spoken language’s grammatical structure would generate an intuitive command model. Furthermore, all of the commercial products reviewed in the literature study followed a similar “predicate -> object” command model. The subject component was left out in favor of either having a non-speech controlled speech affordance or be operating in an always-listening type of fashion.

The Hi-Fi prototype implemented this model for all interactions pertaining to the “Speech for launching and getting answers” scenarios.

4.3.2.3 One-shot dictating

The one-shot dictating mechanic came from the Bing Speech platform and was implemented without any modification to provide the technical speech-to-text transcription functionality. The mechanic entailed that speech transcription was a session based interaction, or “activity”, given its firm separation between different interaction modes. Each transcription session had an explicit beginning and an equally explicit ending. This meant that no

commands could be given during speech transcription since they too would be treated as part of the message to transcribe.

The mechanic was implemented in the “speech for writing” part of the Hi-Fi prototype where it was used as an additional input method for entering text.

4.3.2.4 Context activated dictation

The “context activated dictation” mechanic was conceived during the concept sketching activity of the conceptualization stage. It was loosely based on the multimodality findings of the literature review. The context activation mechanic was about making dictation readily available alongside an existing onscreen touch keyboard whenever a text input field was about to be populated with text.

The “speech for writing” part of the prototype used off-the-shelf Bing Speech UI components to provide a speech transcription feature and GUI. By default, these components relied on a GUI component to initiate a speech transcription session. By implementing the “context activated dictation” mechanic, the standalone “activate speech transcription” button was replaced with an “invisible” contextual trigger.

4.3.2.5 Feedback sounds

The general conception of using sounds for feedback was largely based on findings related to sonic feedback in the literature review. The particular implementation was conceived during concept sketching.

In the prototype, sonic feedback was used to inform users of the listening state of the speech interface. When the interface is attentive, it played a pitch-up sound and when the interface stopped listening, it played a pitch-down sound.

4.4 Hi-Fi prototype

Final stages of the design process included development and user testing of a Hi-Fi prototype. The prototype’s main purpose was to enable user testing of the design proposals that represents the outcome of the conceptualization stage. This resulted in the creation of a prototype that primarily focused on supporting a number of user tasks that were to be given to participants in upcoming user tests. Whereas each of these user tasks were designed to accomplish different tasks, their joint purpose was to provide an interactive environment in which it would be possible to evaluate how the addressing of the proposed speech interface is experienced by users.

4.4.1 Flow chart

The aforementioned interface and interaction components, mentioned in Chapter 4.3, formed the basic building blocks used to create the Hi-Fi prototype. Illustration 22 presents a flow chart on the following page and depicts how these building blocks were implemented in relation to each other inside the prototype speech interface.

The flow chart uses rounded rectangles to depict a system action or user intent while diamonds represents a choice or alternate path. Rectangles indicate an action, event or outcome. Boldly colored rectangles with stripes illustrate interaction mechanics while the pale rectangles with stripes depict interface components.

Green and blue boxes pertain to the “speech for launching and getting answers” parts of the prototype whereas the purple and black boxes pertain to the “speech for writing” parts. Technically, the speech interface is built as two distinct modules which are fitted together closely. These two modules are colored in grey in the flow chart in Illustration 22.

The first module, called “SAPI” in the flow chart, provides the “speech for launching and getting answers” functionality and is implemented as a background service which starts synchronously with the ordinary Windows Shell. Once startup and initiation is complete, the speech input service will display the feedback toast which disposes the speech input affordance to the user.

The default state of the speech input service is to listen for the device’s name and a command to execute. After any successful interaction, this is the state the interface will return to, this state is visible by looking at the two blue diamonds in the flow chart. Accessible from this state are the actions “Ask a question”, “Make a search” and “Open application”. All these actions takes advantage of the “Addressing a device by name” and the “Social command model” interaction mechanics. Furthermore the actions also make consistent use of audible feedback.

The second module, called “Bing Speech” in the flow chart, enables “speech for writing” functionality and is implemented as a standalone Windows RT Application. This module aims to mimic the default mail client application of Windows 8 and then augment it with the speech input affordance. The app is launched either traditionally from the Start screen of Windows 8 or using the functionality provided by the first module of the speech interface. Once the speech enabled mail client is started, the “context activated dictation” mechanic activates speech transcription every time the user clicks or taps on the message text box. Additionally, the mechanic also makes the dictation ribbon to stay active if the user is “writing with speech” and similarly also deactivate it if the user were to use another input method than speech, such as a keyboard. The session based speech interaction mechanic meant that every message had to be transcribed in its entirety without interruption.

The “default Windows components” included UI components such as behaviors and visual elements which were used to create a dictation ribbon that was docked to an onscreen keyboard. Collectively, all these components delivered a speech for writing affordance.

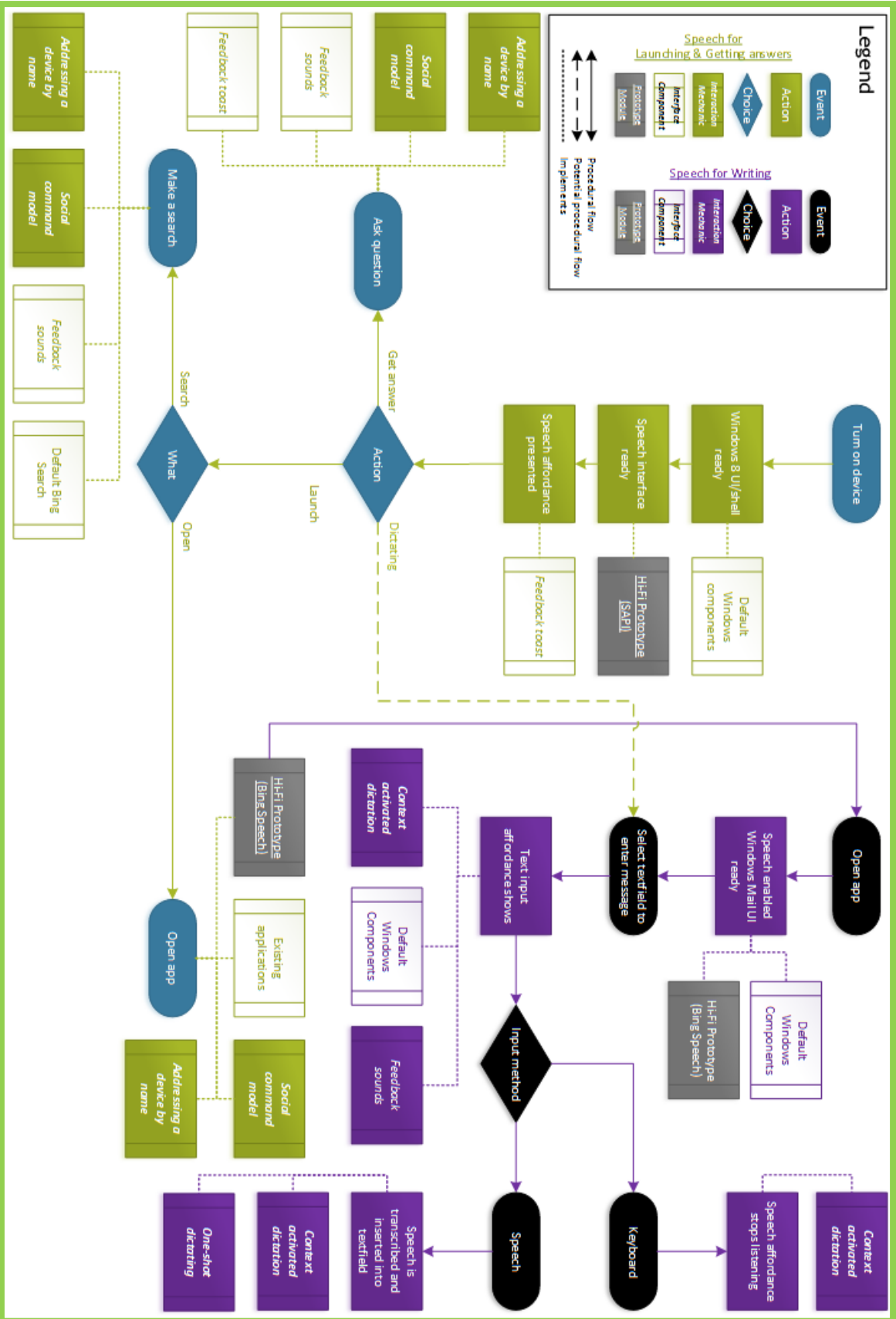


Illustration 18: Flow chart of the Hi-Fi prototype.

4.4.2 Speech for launching

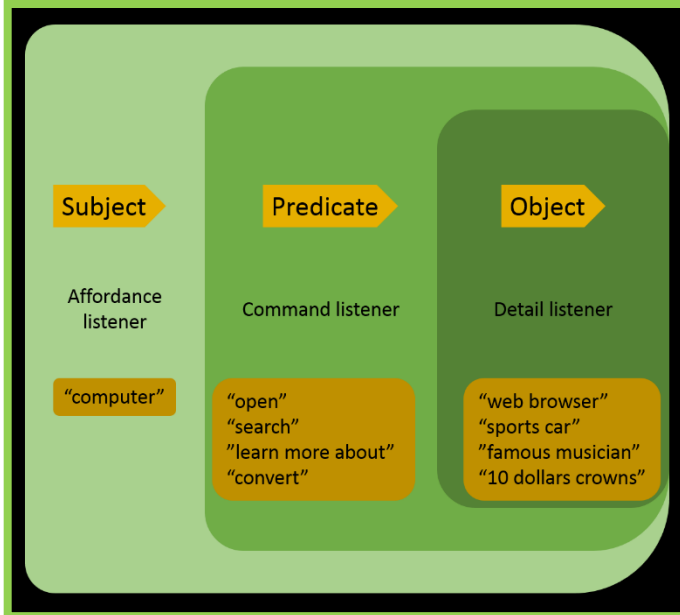
While not explicitly detailed in previous concept sketches, a key functionality of the prototype is the ability to open applications and perform searches universally, regardless of context. To deliver this experience, the conceptual “feedback toast” design was used to inform users about the availability of a speech interaction affordance as well as to instruct them how it can be engaged. As defined in the related concept, this notification will display every time a user logs in on a speech enabled device.

The speech interface was omnipresent and accessible universally from any computer state, and did not rely on a graphical user interface. Instead, the speech input affordance was presented using the aforementioned toast element which was also used to provide limited visual output if needed. Early concepts discussed the inclusion of visual feedback, possibly using the toast element, although later concepts, including the Hi-Fi prototype, opted for sonic feedback in its place.

A recurring theme in this thesis is socially natural. The intent has always been to craft a speech experience that incorporates the humanistic feel of a human-human conversation. This perspective greatly affected the process of designing a command model for interpreting recognized speech. In multiple human languages, including English, a typical sentence is built up by a subject, a predicate and an object.

Consider the sentences “Hampus, write a thesis.” and “Computer, open web browser”. In the first sentence, which was intended for a person, “Hampus” is the subject, “write” is the predicate and “a thesis” is the object. Further on, looking at the second sentence shows an identical structure where “Computer” is the subject, “open” the predicate and “web browser” the object.

As noted above, the social “command model” used in verbal human-human communication was adopted as a reference design for this prototype’s command model. This was a deliberate design decision based on the notion that using common day speech conventions and talking style would help reduce the gap between humans and computers.



The prototype included a custom implementation of the SAPI engine that adhered to the command-model seen in the illustration on the left. With this model the prototype provided users with the ability to open any application on their device by saying the applications name. An example would be: "Computer, open calendar". Similarly, by saying "Computer, search speech recognition" would trigger a search using the operating system's built-in search engine. Based on this feature-set, the prototype could be seen as an omnipresent springboard that launches applications and performs searches using voice input.

Illustration 19: Verbal command model inspired by HHI.

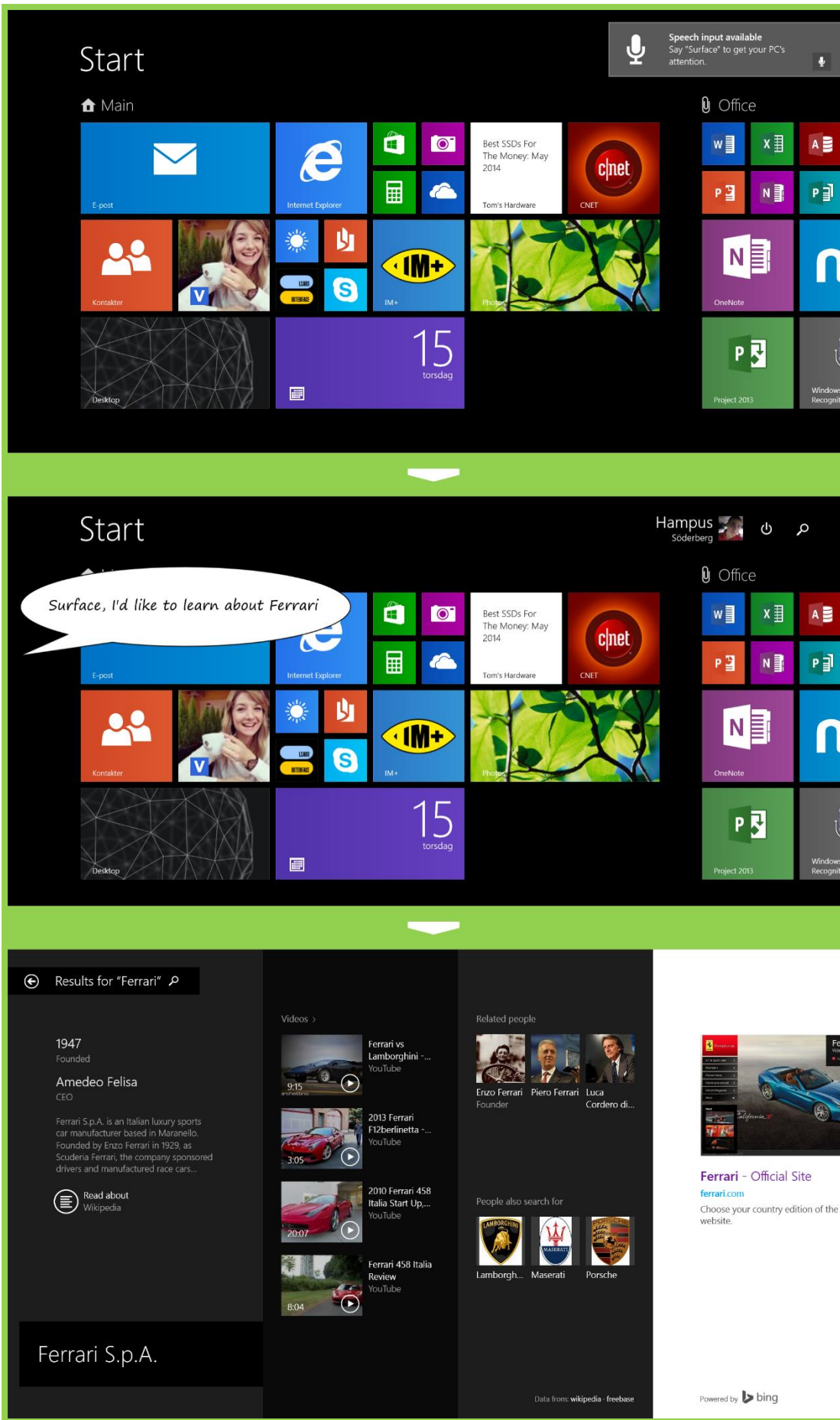


Illustration 20: Hi-Fi prototype - Speech for launching.

4.4.3 Speech for getting answers

The “dynamic action” and “Q&A” concepts were created to demonstrate how speech input may be used as a non-intrusive shortcut for getting short answers to simple questions. Alongside the open and search related functions, the prototype implemented another command, or predicate if you will, for a Q&A task that converts currencies. Being universally available, the Q&A currency converter’s service could be summoned from any context. In Illustration 13, a user is using a web browser when he or she finds a car sales ad which’ price tag is set in Swedish crowns. To convert the price into US dollars, the user gives the computer a command using the subject, predicate and object model. The answer is delivered visually inside a feedback toast that displays the finished conversion. While the prototype uses simulated values, a final product may rely on a third-party web service to do the conversion.

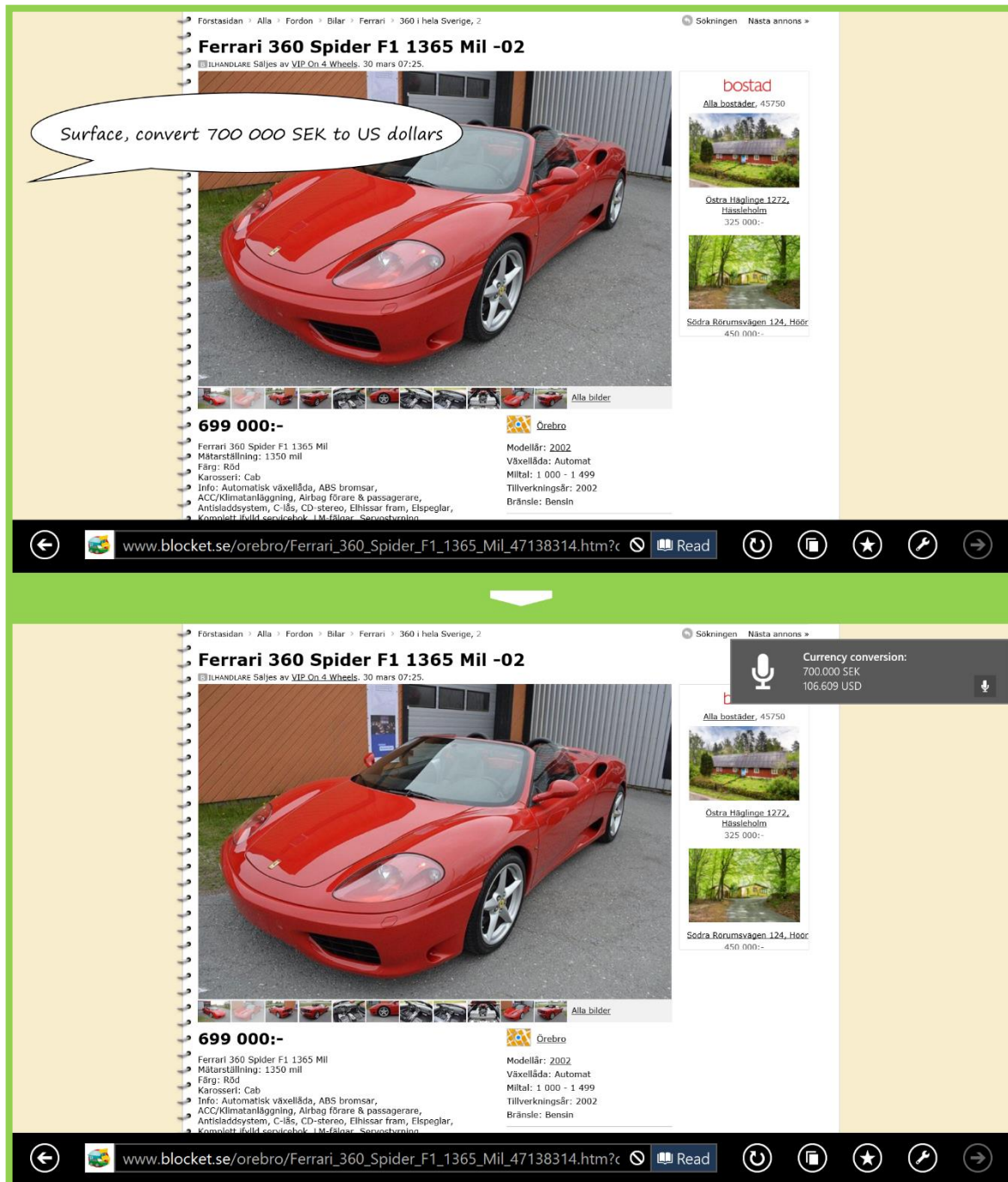


Illustration 21: Hi-Fi prototype - Speech for getting answers.

4.4.4 Speech for writing

This concept was developed separately from the other concepts since it did not revolve around the interpretation and execution of verbal commands. Instead this concept was about “blindly” transcribing speech to text, a process commonly referred to as dictating.

Since the SAPI engine is optimized for recognizing particular words and phrases, Bing speech was chosen for its speech-to-text centered design.

The voice keyboard sketch portrayed a concept where an on-screen touch keyboard was augmented with speech recognition. The focus point was to investigate how two independently redundant input methods could be merged to form a multimodal input affordance which would use the same interaction mechanics to trigger both input methods. In this case, speech input would be triggered along with the integral touch keyboard, making both input methods readily available anytime a text field is selected.

While it would technically remain as two interaction modes, the mode selection would be achieved transparently through listening; if the user speaks, that speech gets transcribed and fed into the selected text field, if the user doesn't speak, nothing gets transcribed and consequently nothing gets fed into the text field apart from text typed on the touch keyboard. This way, users could instantly start speaking or typing their message without having to take explicit action for engaging or disengaging either input modality.

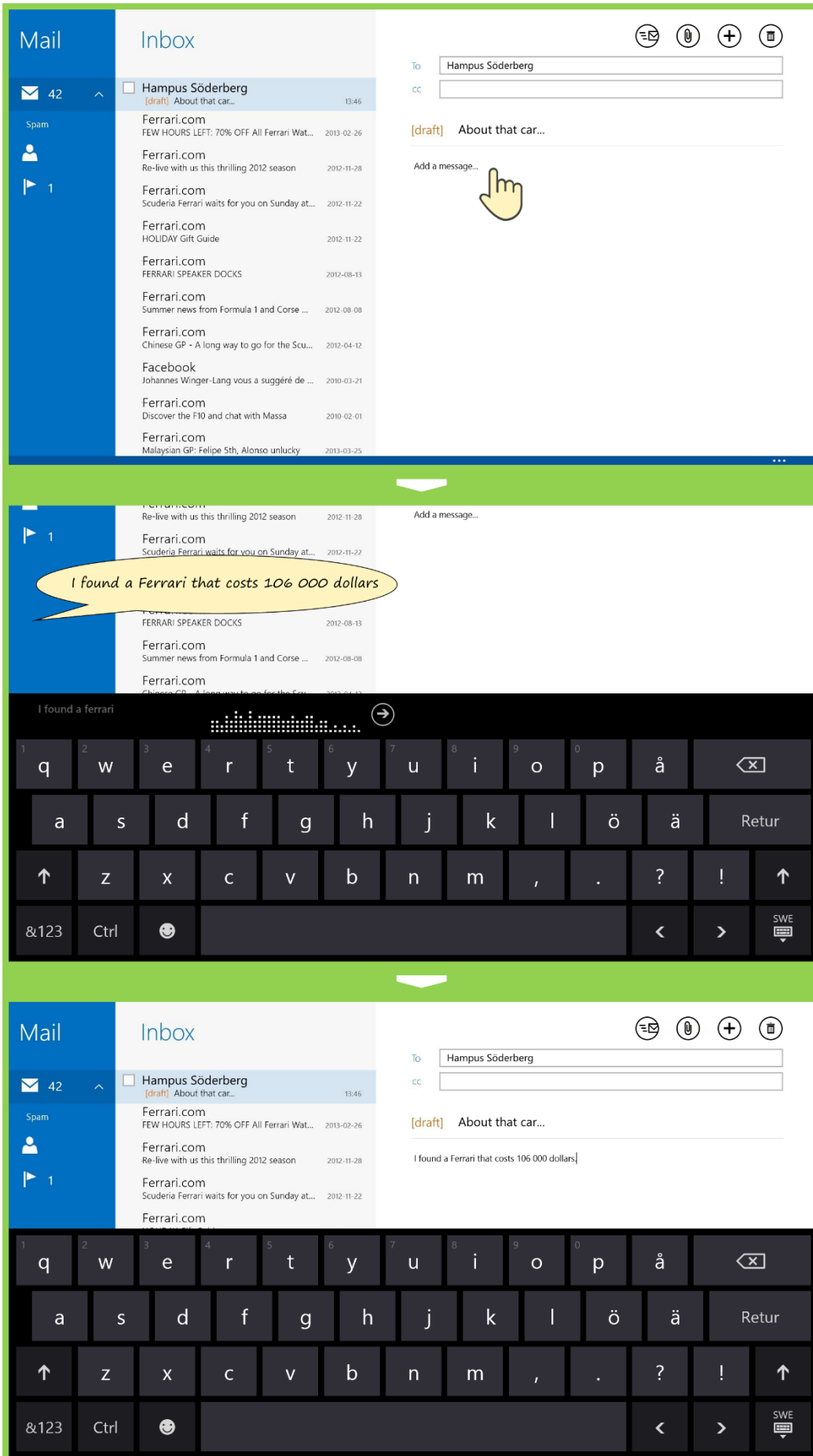


Illustration 22: Hi-Fi prototype - Speech for writing.

4.5 User test

The primary purpose of the user test was to evaluate the proposed speech interface design that was conceived in the conceptualization stage and then realized in the Hi-Fi prototype. Since the Hi-Fi prototype was created solely for this purpose, the user tasks pertaining to the user test were central considerations when designing and developing the prototype. This led to scripted user tests where participants were supposed to do three accumulative tasks. The tasks were thoughtfully planned to expose users to all of the functionalities provided by the proposed speech interface.

User tests were done in the homes of the participants testing the prototype. This context was desired since the environment should be familiar and one where the participants are used to using a personal computing device. Five arbitrarily selected people between 20 and 40 years old were included in the user test. All of the included participants were accustomed to using computers and smart devices as part of their everyday lives. Everyone had at least some past experience of using a speech interface but neither said to make common use of it.

The user test made use of a tablet PC that had the Hi-Fi prototype with the proposed speech interface implemented. Participants were to receive this tablet PC along with a set of tasks to perform (see Appendix: A).

The goal for the user test was to evaluate the following:

- How natural and intuitive the speech affordance is perceived to be.
- How predictable and dependable the speech affordance is perceived to be.
- How useful the speech interface is perceived to be.

Presentation of the speech interaction affordance made sense to most users. It made them understand that the computer would listen for speech commands and it successfully instructed them on how to engage it. After recognizing the computer name, the computer played a pitch-up sound that indicated that the computer was listening and similarly played a pitch down sound when it stopped listening. These sounds were subtle and found to be mistakenly similar and did not serve their purpose well in any of the user tests. They were either not noticed, not differentiated or interpreted as unrelated to the speech interface. Despite the previous predicament, the interaction mechanic in itself was perceived by users to make sense both socially and from a dependability perspective. It was not immediately understood by everyone, but as soon as they did understand it, they found the social command model to be both logical and natural. One test participant even went so far as to say: *“Addressing the computer by its name makes it feel human and would probably work well in situations where multiple voice controlled computers exist near each other.”*

The aspect of multimodality was explored through the extension of a touch first user interface. Generally, users had no difficulties moving between the two input methods apart from a preference towards using speech which was possibly caused by the imposed situation of testing a speech interface.

The currency conversion feature was highly appreciated for its context-preserving behavior and rapid invocation. The fact that it understood acronyms was particularly valued by one user who pointed out that he or she may not always know the correct pronunciation of a foreign currency and would thereby prefer to speak its acronym instead.

Additionally, the model of having the speech affordance ubiquitous was observed to be a

pleasant experience for all test participants as they could easily see themselves using speech as a quick way to access out-of-focus or peripheral applications and services.

The closest integration of speech and touch was made in the “voice keyboard”-equipped mail application. Consistently observable throughout every user test was the desire to give app-specific commands. Either directly through speaking the name of buttons and other interactive elements or through issuing activity centered commands, such as “Computer, write an email”. This had a particular impact on the mail app since it forced users to use touch to create and initiate the dictating affordance.

Overlooking the misunderstandings related to being forced into using touch, the automatic invocation of speech was received with mixed perceptions. Some participants enjoyed the fact that he or she didn’t have to do any extra actions before using the speech dictation, while others felt like they were losing control.

Common for every participant was the annoyance of not being able to pause or edit specific words or sentences when dictating their messages. This annoyance was experienced by all participants who were all disappointed by this behavior. One participant exclaimed: *“That’s strange... It must not stop transcribing just because I pause to think.”*

Similarly, having to use touch to re-focus the message body text field and consequently re-engaging the speech keyboard was perceived as clumsy. Additionally, the behavior of replacing an already transcribed message with a new one made the experience even worse.

Beyond the act of addressing the interface, all participants found it difficult to figure out what they should say to the computer. The related interview question indicated a wish for an introductory tutorial or command reference sheet to prevent this from becoming an issue. The most common “mistake” the participants made was to skip the command predicate and instead speak only the computer name directly followed by the application to launch.

5. Summary and analysis

5.1 Summary

The central point of interest revolved around investigating how a speech interface should be addressed or engaged to get its attention. The intention was to design the speech interface's interaction affordance in such a manner that it would be perceived as socially natural to use. The speech transcription affordance should also be readily available when needed and effortlessly discarded when not.

Sketching, prototyping and user tests were used as tools for conceptualizing and evaluating potential design solutions. A literature review was conducted prior to the conceptualization and implementation stages to deepen the understanding and knowledge about the research area and to learn about current and past efforts.

Although the outcome of the thesis was accumulated throughout every stage of the research project, the research goals and research questions were particularly investigated during user testing of the Hi-Fi prototype. The prototype included a design proposal that attempted to fill the identified design gap related to addressing and starting a speech interaction.

Furthermore, the Hi-Fi prototype was designed and developed to add speech input alongside traditional touch input which resulted in a multimodal user interface and a multimodal interaction experience for users. This was motivated by two factors. Firstly, to create an approachable use context that most people were able to relate to, extending an existing touch interface with additional speech features, seemed rational as it provided a familiar environment with familiar tasks. Secondly, the thesis is done from an input method agnostic perspective where it is believed that every interaction method brings different characteristics that may become useful in different situations and contexts.

By creating a multimodal touch and speech enabled prototype it was possible to investigate not only how a speech interaction should begin, but also *when* a speech interaction may make sense as well as *how* its mechanics should be designed. To remain focused on the matter of interaction mechanics related to the activation, addressing and presenting a speech affordance, visual and audial interface elements were based on default design patterns of Windows 8.

5.2 Speech for launching & getting answers

The Hi-Fi prototype provided ubiquitous access to speech input regardless of the current activity or open application. Interaction mechanics of the speech interface were designed to rely on audial components to explore what possibilities and limitations that that may impose on the perceived usability of the speech interface. The most prominent sign of this was the absence of a dedicated GUI for the speech interface. Early sketches illustrated a concept where presentation of the speech input affordance was absent as well. Revisions of the concept later recognized that since speech input is not commonly implemented in a similar way, it would presumably be very hard or impossible for a user to intuitively assume that

speech input was available. With that motivation it was concluded that while it may work well in a future where speech interaction is well-established and often taken for granted, the current situation, where speech interaction is being only sparsely available, requires the input method to be presented in some way.

5.2.1 Presentation

The speech interaction affordance was presented using a pop-up notification, referred to as a feedback toast. This toast would be displayed shortly after a user login to inform the user that the computer supports speech interaction. Additionally, the toast would also instruct the user to say the computer's name to get its attention.

The notification toast served its purpose well in terms of getting people to start talking to the computer. Upon receiving the first user task to solve using speech, participants of the user tests would often find themselves a bit insecure of what they were supposed to say or do, but once the toast appeared they would follow its instructions and successfully engage the speech interface.

5.2.2 Activation

The speech user interface's equivalent of an activation button was to say the computer's name. Once the speech interface recognized the computer name it would start listening and processing speech for a short period of time. After this period, the interface would return to its previous state of listening only for the computer name.

Using the computer's name as the keyword was found to add a sense of humanistic feel to the human-computer interaction dialog. User testing revealed that all participants thought that it made sense to say the computers' name and some reflected that that's how they speak to real people.

Having a limited time to speak did impose difficulties speaking long commands which caused participants to stop and think about sentence structure and pronunciation to be able to speak the command as quickly as possible. Furthermore there were major issues with users falling out of sync which will be discussed in more detail in the following section (Chapter 6.2.3). Having to say the computer's name before each speech interaction did not seem like an annoyance and thus the concept of having the interface not listening until it heard its name did not appear to be cumbersome.

5.2.3 Sonic feedback

To inform users whether or not the speech interaction affordance was enabled, i.e. whether the computer listens, feedback in the form of a pitch-up and a pitch-down sound was provided. The pitch-up sound was played when the speech interface recognized the computer name and started listening for speech input. The pitch-down sound played when the interface stopped listening, either because interaction grace period ended or the speech input was

understood.

These sounds did fulfill their purpose since they did not provide understandable and reassuring feedback to the user. Not everyone noticed the sounds, and when they did, they had trouble distinguishing the two from each other or would assume that the sounds didn't belong to the speech interface.

This created a lot of confusion since users didn't know when the device was listening and would consequently end up being out of sync and having difficulties issuing commands.

5.2.4 Visual response

In addition to presenting the speech interaction affordance, the “feedback toast” design was also used to deliver visual answers to quick verbal questions. The toast notification itself is moderately small and is thereby unable to provide detailed answers. The upside to the small size is that it doesn't force the user to leave context.

Generally the visual response was highly appreciated for its context sensitive behavior and native visual design. Participants did however express concern about it possibly being too small and not feature enough detail to be useful in some situations.

5.2.5 Analysis

Using the feedback toast to inform users that the computer was listening for voice input worked as intended. Users were made aware that the speech input was already available and did not require any pre-steps to get started. Additionally its instructions were delivered successfully since users, after seeing the toast, instantly said the computer's name to get its attention. While this provided a guided first-use experience that was well perceived, to continue and make a successful interaction, users had to formulate a speech command consisting of a subject, the computer name, a predicate and an object. While everyone found this command model to be logical and feel socially natural, they would have needed guidance from the speech interface to learn how to speak to it.

Using a computer's name to get its attention felt natural to all participants of the user test. Some minor concerns did however surface; the computer name needs to be short and easy to pronounce, yet unique and hard to misinterpret.

The sounds used for indicating attention status of the speech interface did not fulfill their function. Instead of providing solid feedback, the sounds were unheard or misinterpreted, causing an almost chaotic experience where users would try to get the computers attention when it was already listening and trying to speak commands when the “stop listening”-sound was heard.

A possible explanation to their lack of usefulness may be that the sounds were gentle and sounded similar. This would indicate that more prominent and distinguishable sounds could help solve the problem.

However, a different solution, could be to include visual feedback to further indicate if the

interface is listening or not. A similar solution was explored in the first round of concept sketching (see Chapter 5.1.1).

The inline context preserving notification toast was perceived to be efficient and respectful, presumably because of its concise text message and small size. Although it was already observed to be an appreciated function and found to convey an enjoyable experience, some users suggested that it could reply by voice instead of the toast or provide a bigger toast providing more detail.

5.3 Speech for Writing

Part of the ubiquitous access model and exploration of multimodality was the “voice keyboard” concept. Even though it was conceptualized and conceived in tandem with the other concepts, the voice keyboard did not make use of the same interaction mechanics as other parts of the proposed speech interface. The major difference was related to the difference between ubiquitously available and context-driven invocation. Whereas the user could always access the speech interface by calling out the computer’s name followed by a command, the voice keyboard was only accessible when a text box was focused. While the concept defined this to any text field, the Hi-Fi prototype only implemented this functionality inside a voice enabled e-mail application.

By not adhering to the same interaction mechanics as the rest of the prototype, the voice keyboard created an opportunity to evaluate another activation model that is based on context rather than explicit user engagement.

5.3.1 Presentation

The voice keyboard could only be seen after it had been activated. Once activated, it would integrate itself with the touch keyboard as a thin ribbon above it. The ribbon showed a VU-meter that visualizes what sounds the computer hears as well as a streaming textual representation of the recognized speech.

The dictation bar efficiently conveyed that the computer was listening for spoken input but it was not as efficient at informing users that it would transcribe the users’ speech.

5.3.2 Activation

In contrast to the universally available command interface, the voice keyboard could not be invoked manually by a spoken command. Instead the voice keyboard was activated in the same way that the default Windows 8 touch keyboard was, namely by giving focus to a text input field.

If it was understood that the app was not to be controlled using voice, the invocation method worked well for engaging the dictation affordance. However since re-activating the voice keyboard required the user to perform the same activation process again, the user had to first

de-select and then re-select the text field which was found to be both cumbersome and distracting by test participants. Furthermore, upon successfully reactivating the voice keyboard and transcribing another piece of speech, it would replace the previous transcription. To add even more frustration to the situation, the voice keyboard automatically stopped listening after a moment of silence which made users experience a lot of unintentional interruptions and forced them to start over from the beginning.

5.3.3 Sonic feedback

Similar to the command affordance, whenever the voice keyboard started or stopped listening for speech to transcribe, it issued a short pitch-up or pitch-down sound to indicate whether it was listening or not.

Comparable to the command affordance sounds, these sounds did not function adequately either. Apart from poor sound design which has been covered previously, a particular reason for these sounds might be because they did not align with the other feedback sounds, neither in tune nor in timing.

5.3.4 Visual response

Post-invocation, the voice keyboard affordance would disappear and leave the user with only the default Windows 8 touch keyboard. This would happen after every activation, including successful recognition or the user not speaking at all and instead start touch typing the message. After dictating, the transcribed speech would instantly appear in the related text field, replacing any existing text. While this constitutes feedback delivered post interaction, the speech transcription bar above the touch keyboard also provided continuous feedback for each recognized word.

The continuous feedback was observed in user tests to provide guidance related to making people aware of what the computer heard and could thereby choose to either continue speaking or start over if too many misrecognitions were made. Instantly putting the transcribed speech into the selected text field was appreciated since it helped make the voice keyboard feel rapid.

5.3.5 Analysis

The thin dictation ribbon above the keyboard successfully informed users that their voices were listened to. What it did not convey was the instruction that he or she should speak the whole message at once without stopping.

If this information was made available, users' might have perceived the interface as less awkward because they would at least have been aware that that's how it behaves. The current implementation caused frustration and confusing among users since they were unable to edit or add to a transcribed message.

Making the dictating interface appear together with the touch keyboard whenever a text input field was selected made sense to users, if they understood that they could not use speech to start dictating. What can be derived from this is that users want to be able to use speech to perform in-app commands both for pushing buttons but also to engage in an activity, in this case, writing an email.

Even if the first activation of the dictating interface was successful, users were repeatedly observed to try to reactivate it by issuing voice commands. The wish to reactivate the dictating interface was a result from the interface stopping to listen before the user had finished speaking his or her message. During user testing, adding a touch button to re-activate the voice keyboard came up as a possible solution.

Similar to previous findings, the audible feedback was not very clear for the voice keyboard either. It did however not affect the overall user experience in the same magnitude as it did in the “speech for launching and getting answers” part of the prototype. This could possibly be because the dictating affordance implemented in the voice keyboard also offered visual feedback in the speech transcription bar.

If more descriptive sounds were used and if those corresponded and were synced with the command affordance’s sounds, the audible feedback of both might have made more sense and have been able to provide the guidance and increased confidence that they were originally intended to bring.

Having visual feedback improved users’ confidence in the speech transcription accuracy. By automatically inserting the recognized speech into the designated text field, the speech interface was perceived as seamless and quick to use.

6. Conclusion and discussion

Throughout the thesis project, the overall purpose has been to contribute additional knowledge about fundamental interaction mechanics that needs to be understood in order for designers to be able to design useful and intuitive speech interfaces for human-computer interaction. The focus was narrowed to investigating the mechanics of how to engage a speech interface. To be able to investigate these mechanics they had to be put into context. For this reason, a number of scenarios were created and explored throughout the design process. While the focus was set on the interaction mechanics, the individual scenarios were to some extent also evaluated in the user test.

6.1 Conclusion

The outcome of this project has made it possible to answer both research questions as well as to provide additional insight into other aspects pertaining to the design of a speech interface.

1. *Using common HHI patterns, how can a speech interface be designed to be perceived as socially natural to engage a speech interaction with?*

By implementing a social command model, subject -> predicate -> object, a speech interface can be perceived as socially natural. Furthermore, by making the “subject” component of the social command model correspond to the name of the target device, a speech interface is able to seamlessly start listening whenever a user wants to interact with it.

2. *How can a speech dictation affordance be integrated into a multimodal touch and speech input interface to offer seamless activation?*

By making use of the same activation mechanics that touch based text input methods do, it is possible to integrate a seamless “initial activation” mechanic of a speech dictation affordance in a multimodal touch and speech input interface.

6.2 Discussion

Meanwhile this thesis' focus was set on investigating the mechanics of how a speech interaction should be initiated, other topics of relevance to the field of interaction design were also investigated. The most prominent of these topics were multimodality and feedback.

By borrowing GUI elements originally crafted for a touch input interface, speech input was able to produce visual output that was consistent to the output produced by the touch input method. In this way, it was possible to create interaction mechanics specifically tailored for a speech input interface while maintaining visual continuity and behavioral coherency with an existing touch optimized UI.

Employing audio as the only feedback did not provide users with enough confidence to enable fluid speech interaction. Since feedback was not the main focus and audio design was outside the scope of this thesis, it is believed that if a thorough audio design process would've been conducted, those resulting sounds might have generated a different outcome.

The dictation part of the prototype provided a mix of audible and visual feedback which was found to induce more confidence than the parts using only sonic feedback.

The social command model provided a natural way for users to initiate an interaction for launching an application or performing an action. Its structure and utility was found intuitive and useful. When evaluated, it generated an important research finding for future investigations of speech interaction interfaces.

It was discovered that users subconsciously expect to be able to control everything using speech. This would include being able to activate GUI affordances by speaking their corresponding names or giving semantically more complex speech commands that instructs the computer to carry out a task rather than merely launching an application. The "speech for getting answers" concept employs some aspects of this by using a *convert* predicate instead of the "speech for launching" concept's *open* predicate.

By implementing a speech transcription or dictating functionality in a session based manner where the only interaction mechanics available are start and stop, users are forced to speak their entire message without the ability to stop and think. Once a message was transcribed, the user could either accept it or start over with an empty message.

This behavior induced anxiety and frustration in test participants who could not fathom how they were supposed to know exactly, word-for-word, what they wanted to write beforehand. Conclusions which can be made from this are that people have to be able to think in their own rhythm and they also need to be able to edit what they've previously said.

6.3 Future work

Speech interaction has long been regarded to have the potential of being a truly natural and efficient interaction method. In this thesis' effort of investigating how to address such an interface, a few areas for future research was discovered and identified.

- Investigate how to integrate computer naming during out-of-box experiences with the design of a speech interface.
- How to provide introductory instructions on how to speak and formulate commands to the interface.
- Explore how synthesized voices (speech-to-text) can be used as output.
- Investigate how editing capabilities, comparable to those found in a typical GUI application, should be integrated in a speech transcription interface.
- Further investigate the interaction mechanics of speech transcription interfaces.
- Investigate how the social command model can be extended to enable more complex command structures and support more usage scenarios.

7. References

- Absar, Rafa, and Catherine Guastavino. "USABILITY OF NON-SPEECH SOUNDS IN USER RINTERFACES." *Proceedings of the 14th International Conference on Auditory Display*. Paris, France: ICAD, 2008.
- Apple. *Apple - iOS 7 - Siri*. 2013. <http://www.apple.com/ios/siri/> (accessed 03 22, 2014).
- . *Apple Human Interface Guidelines: The Apple Desktop Interface*. Addison-Wesley, 1987.
- . *Apple Reinvents the Phone with iPhone*. 01 09, 2007. <http://www.apple.com/se/pr/library/2007/01/09Apple-Reinvents-the-Phone-with-iPhone.html> (accessed 03 15, 2014).
- Baker, James K. "The DRAGON System - An Overview." *TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING*. Yorktown Heights, NY, USA : IEEE, 1975. 24-29.
- Bakker, Saskia, Elise van den Hoven, and Berry Eggen. "Knowing by ear: leveraging human attention abilities in interaction design." *Journal on Multimodal User Interfaces* (Springerlink), 2012: 3-4.
- Bellotti, Victoria, Maribeth Back, W. Keith Edwards, Rebecca E Grinter, Austin Henderson, and Cristina Lopes. "Making Sense of Sensing Systems: Five Questions for Designers and Researchers." *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (ACM), 2002: 415-422.
- Brewster, Stephen, et al. "We Need to Talk: Rediscovering Audio for Universal Access." *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*. Stockholm, Sweden: ACM, 2011. 715-716.
- Brewster, Steven A. *The Design of a Sonically-Enhanced Interface Toolkit*. The University of Glasgow, n.d.
- Buxton, Bill. *Multi-Touch Systems that I Have Known and Loved*. 2007. <http://billbuxton.com/multitouchOverview.html> (accessed 03 06, 2014).
- . *Sketching User Experiences: Getting the design right and the right design*. USA: Elsevier / Morgan Kaufman, 2007.
- Carrol, John M. "Five Reasons for Scenario-based Design." *Hawaii International Conference on System Sciences*. Blacksburg, VA: IEEE, 1999.
- Hoste, Lode, and Beat Signer. "SpeeG2: A Speech- and Gesture-based Interface for Efficient Controller-free Text Entry." *International Conference on Multimodal Interaction - ICMI*. Sydney, Australia: ACM, 2013.
- Kalnikaitė, Vaiva, Patrick Ehlen, and Steve Whittaker. "Markup as you talk: establishing effective memory cues while." *CSCW'12*. Seattle, WA, USA: ACM, 2012. 349-358.
- Karat, John. "Scenario Use in the Design of a Speech Recognition System." In *Scenario-based Design*, by Carrol and John M., 109-133. New York, NY, USA: ACM, 1995.

- Lauesen, Soren. "More on usability testing." In *User Interface Design: A Software Engineering Perspective*, by Soren Lauesen, 413-442. Harlow, England: Pearson/Addison-Wesley, 2005.
- Mark, Weiser, and John Seely Brown. *THE COMING AGE OF CALM TECHNOLOGY*. Xerox PARC, 1996.
- Microsoft. *Bing Developer Center: Speech*. n.d. <http://www.bing.com/dev/en-us/speech> (accessed 03 20, 2014).
- Microsoft Corporation. *Microsoft Academic Search*. 2014. <http://academic.research.microsoft.com/> (accessed 2014).
- Microsoft. *Guidelines for common user interactions (Windows Store Apps)*. 06 21, 2013. http://msdn.microsoft.com/en-us/library/windows/apps/hh465370.aspx#design_for_a_touch-first_experience (accessed 03 16, 2014).
- *Speech API Overview (SAPI 5.4)*. 2009. <http://msdn.microsoft.com/en-us/library/ee125077%28v=vs.85%29.aspx> (accessed 03 20, 2014).
 - *Use Speech on my phone | Windows Phone How to (United States)*. 2012. <http://www.windowsphone.com/en-US/how-to/wp8/basics/use-speech-on-my-phone> (accessed 03 20, 2014).
 - *Windows Vista Demo: Windows Speech Recognition*. 2006. <http://www.microsoft.com/enable/demos/windowsvista/speechdemo.aspx> (accessed 03 20, 2014).
- Moggridge, Bill. *Designing Interactions*. Massachusetts: MIT Press, 2007.
- Morency, Louis-Phillipe. "Modeling Human Communication Dynamics [Social Sciences]." *IEEE Signal Processing Magazine* (IEEE) 27 (2010): 112-116.
- Morency, Louis-Phillipe. "Modeling Human Communication Dynamics [Social Sciences]." *IEEE Signal Processing Magazine* (IEEE) 27 (2010): 112-116.
- Mott, Tim, interview by Bill Moggridge. *The Desktop (Office) Metaphor* (September 1, 2003).
- Mott, Tim, interview by Bill Moggridge. *The Desktop (Office) Metaphor* (September 1, 2003).
- Mustaquim, Moyeen Mohammad. "Automatic speech recognition: an approach for designin inclusive games." *Multimedia tools and applications*, 2013: 131-146.
- Nielsen, Jakob. "How Many Test Users in a Usability Study?" June 04, 2012. <http://www.nngroup.com/articles/how-many-test-users/> (accessed June 30, 2014).
- "Why You Only Need to Test With 5 Users." *Nielsen Norman Group: UX Training, Consulting & Research*. March 19, 2000. <http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/> (accessed June 30, 2014).
- Nielsen, Jakob, and Thomas K. Landauer. "A Mathematical Model of the Finding of Usability Problems." *INTERCHI'93*. NJ, USA: ACM, 1993. 206-213.

- Noessel, Christopher, and Nathan Shedroff. *Make it so - Interaction Design lessons from Science Fiction*. Brooklyn, NY, USA: Rosenfeld Media, LLC, 2012.
- Norman, Donald A. *The design of everyday things*. London: MIT Press, 1998.
- Nuance. *Nuance*. 2014. <http://www.nuance.com/index.htm> (accessed 03 20, 2014).
- Nudelman, Greg. *Android Design Patterns: Interaction Design Solutions For Developers*. Indianapolis: John Wiley & Sons, Inc, 2013.
- Star Trek: The Next Generation*. Directed by Gene Roddenberry. Performed by Paramount Pictures. 1987.
- Rubin, Jeffrey, and Dana Chisnell. *Handbook of Usability Testing: How to plan, design, and conduct effective tests*. Indianapolis, IN: Wiley Pub, 2008.
- Saffer, Dan. *Designing for interaction*. Berkeley, CA, US: New Riders in association with AIGA Design Press, 2007.
- . *Designing Gestural Interfaces*. Canada: O'Reilly Media, 2009.
- Schreder, Günther, Karin Siebenhandl, Eva Mayr, and Michael Smuc. "The ticket machine challenge? Social inclusion by barrier-free ticket vending machines." *The proceedings of the good the bad and the challenging conference*. University Krems, 2009.
- Sharp, Helen, Yvonne Rogers, and Jennifer Preece. *Interaction design: beyond human-computer interaction*. United Kingdom: Wiley, 2007.
- Thurrot, Paul. *Windows XP Tablet PC Edition 2005 Review*. 08 30, 2004. <http://winsupersite.com/article/windows-xp2/windows-xp-tablet-pc-edition-2005-review-127414> (accessed 03 20, 2014).
- Wildstrom, Steve. *Nuance Exec on iPhone 4S, Siri, and the Future of Speech Tech.pinions - Perspective, Insight, Analysis*. 10 10, 2011. <http://techpinions.com/nuance-exec-on-iphone-4s-siri-and-the-future-of-speech/3307> (accessed 03 22, 2014).
- Yuan, Jiahong, Mark Liberman, and Christopher Cieri. "Towards an integrated understanding of speaking rate in conversation." *Proceedings of Interspeech 2006, 9th International on Spoken Language Processing*. Pittsburg, PA, USA, 2006. 541-544.

Appendix A: Test plan

1. Purpose and goals

The thesis aims to investigate how a speech interface can be designed to be dependable and accessible by taking cues from HHI. This user test will be conducted to investigate the proposed solution to this design gap. The first goal is to better understand how the chosen speech affordance design is perceived by a potential user. The second goal is to evaluate how predictable it behaves regarding whether or not it listens to the user.

- How natural and intuitive the speech affordance is perceived to be.
- How predictable and dependable the speech affordance is perceived to be.
- How useful the speech interface is perceived to be.

2. Interview questions

1. Have you ever talked to a computer before?
2. Do you commonly use speech interaction today?
3. Why do/don't you use speech interaction today?
4. What is your first impression of using this speech interface?
5. What was your best experience using the speech interface?
6. What was your worst experience using the speech interface?
7. What do you think about having to say the computer's name every time you wanted its attention?
8. Did the interface seem dependable?
 - a. Would you trust the interface to be attentive once you need it?
 - b. Would you trust the interface to remain passive when you don't use it?
9. Did the speech interface seem like an efficient shortcut or could you just as well have used ordinary touch or mouse/keyboard input?
10. Did the interface seem like an efficient shortcut or did it appear clumsy to use?
11. Did you miss having a visual "target" to look at when talking?
12. After this experience, has your opinion about using speech as an input method change?
13. Would you like to have a speech interface on your device?

3. Test layout

To focus on and be able to evaluate the addressability aspect of the proposed speech interface, the user test will consist of a series of pre-defined user tasks that are derived from the conceptualization stage of the design process.

Information is gathered mainly through observation as the participant will be asked to "think aloud" to give better insight into their experiences. A two-part semi-structured interview is conducted to "break the ice" and collect the participant's current experience with speech input. The second part is done in the end of the user test to trigger after-thoughts and

reflections of the participant. Furthermore, test participants are encouraged to think aloud and share any opinions, ideas or suggestions they might have.

3.1 Presentation

The participant will receive a quick introductory description of the research project to become better prepared for the coming test. Furthermore, the participant will be informed about the estimated duration of the test and that his or her actions and experiences will be documented as written notes by the test moderator.

3.2 User tasks

The user test begins with the participant being introduced into a fictive context where he or she wants to research an Italian sports car, lookup its price and send their findings via email to the test moderator.

1. Using voice input, the user should ask his computer for information about an Italian sports car.
2. Using voice or touch input, the user should find a price tag for such a car.
 - a. Once a Swedish price tag is found, the price should be converted into US dollars.
3. Using voice input, the user should “write” an email to the test moderator where his or her findings are described.

4. Test environment

User tests are conducted in everyday environments where people typically use personal computing devices. The participant will receive an Internet-connected tablet PC pre-loaded with the speech recognition interface affordance prototype.

5. Test moderator role

The primary functions of the test moderator are to observe the participant's actions and experiences as well as to lead the test by giving the participant user tasks to perform. The test moderator will also conduct a short interview before and after the actual test.