Context-dependent voice commands in spoken dialogue systems for home environments

A study on the effect of introducing context-dependent voice commands to a spoken dialogue system for home environments

KARL DAHLGREN

Master’s Degree Project
Stockholm, Sweden 2013
Context-dependent voice commands in spoken dialogue systems for home environments

A study on the effect of introducing context-dependent voice commands to a spoken dialogue system for home environments

KARL DAHLGREN

Stockholm 2013

Master’s at ICT
Supervisor: Fredrik Kilander
Examiner: Fredrik Kilander
Abstract

This thesis aims to investigate the effect context could have to interaction between a user and a spoken dialogue system. It was assumed that using context-dependent voice commands instead of absolute semantic voice commands would make the dialogue more natural and also increase the usability. This thesis also investigate if introducing context could affect the user’s privacy and if it could expose a threat for the user from a user perspective. Based on an extended literature review of spoken dialogue system, voice recognition, ambient intelligence, human-computer interaction and privacy, a spoken dialogue system was designed and implemented to test the assumption. The test study included two steps: experiment and interview. The participants conducted the different scenarios where a spoken dialogue system could be used with both context-dependent commands and absolute semantic commands. Based on these studies, qualitative results regarding natural, usability and privacy validated the authors hypothesis to some extent. The results indicated that the interaction between users and spoken dialogue systems was more natural and increased the usability when using context. The participants did not feel more monitored by the spoken dialogue system when using context. Some participants stated that there could be a theoretical privacy issues, but only if the security measurements were not met. The paper concludes with suggestions for future work in the scientific area.

Keywords: Spoken dialogue system, Context-dependency, Absolute semantic, Ubiquitous Computing, Ambient Intelligence, Voice recognition, Speech communication
Sammanfattning


Nyckelord: Spoken dialogue system, Context-dependency, Absolute semantic, Ubiquitous Computing, Ambient Intelligence, Voice recognition, Speech communication
Acknowledgements

I would like to express my gratitude to my supervisor, Fredrik Kilander, for the useful comments, remarks and engagement throughout the process of writing this master thesis. Furthermore, I would like to thank both of my mentors, Jozef Swiatycki and Henrik Bergström, for always being inspiring and encouraging. Also, I like to thank the participants, who have willingly shared their precious time during the process of interviewing. Mostly, I would like to thank my wonderful girlfriend, Annicka Lundin, who has supported and motivated me throughout the entire process. All of you have my deepest and most sincere gratitude.

I am grateful to the following for permission to reproduce copyright material:

Figures

Figure 2.3 adapted from Paliwal, K and Yao, K. (2010) 'Robust Speech Recognition Under Noisy Ambient Conditions', 'Human-Centric Interfaces for Ambient Intelligence', 1st edition, pp. 138, Figure 6.2, ©Elsevier Inc. 2010, reproduced by permission; Figure 2.1 adapted from McTear, M. (2010) 'The Role of Spoken Dialogue in User-Environment Interaction', 'Human-Centric Interfaces for Ambient Intelligence', 1st edition, pp. 232, Figure 9.1, ©Elsevier Inc. 2010, reproduced by permission; Figure 3.1 adapted from Benyon, D. (2010) 'Designing Interactive Systems: A Comprehensive Guide to HCI and Interaction Design', 2nd edition, pp. 252, Figure 11.1, ©Pearson Education Limited 2005, 2010, reproduced by permission.

Text

Box 2.2 quoted from the movie Iron Man (2008), Paramount Pictures, ©Marvel Studios.
## Acronyms and Abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AmI</td>
<td>Ambient Intelligence</td>
</tr>
<tr>
<td>ASR</td>
<td>Automatic speech recognizer</td>
</tr>
<tr>
<td>DM</td>
<td>Dialog manager</td>
</tr>
<tr>
<td>HCI</td>
<td>Human-computer interaction</td>
</tr>
<tr>
<td>IUI</td>
<td>Intelligent User Interfaces</td>
</tr>
<tr>
<td>UbiComp</td>
<td>Ubiquitous Computing</td>
</tr>
<tr>
<td>PACT</td>
<td>People, Activities, Context and Technologies</td>
</tr>
<tr>
<td>RG</td>
<td>Response Generator</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>SLU</td>
<td>Spoken language understanding</td>
</tr>
</tbody>
</table>
Contents

Abstract I

Sammanfattning III

Acknowledgements V

Acronyms and Abbreviations VII

List of Figures XIII

1 Introduction 1
  1.1 Overview ........................................... 1
  1.2 Problem statement ................................. 2
  1.3 Hypothesis .......................................... 4
  1.4 Purpose ............................................ 4
  1.5 Impact ............................................ 4
  1.6 Delimitations ...................................... 4
  1.7 Outline ........................................... 5

2 Background 7
  2.1 Spoken dialogue systems ......................... 7
  2.2 Speech recognition ................................... 10
    2.2.1 Problems with speech recognition ............... 11
  2.3 Types of commands ............................... 12
    2.3.1 Absolute semantic commands ................... 12
    2.3.2 Context-dependent commands ................. 13
  2.4 Ambient Intelligence ............................. 13
  2.5 Human-Computer interaction ...................... 14
    2.5.1 Usability ................................... 15
    2.5.2 Human-centric design ....................... 16
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 Test design</td>
<td>47</td>
</tr>
<tr>
<td>6.1 Setup</td>
<td>47</td>
</tr>
<tr>
<td>6.2 Scenarios</td>
<td>48</td>
</tr>
<tr>
<td>6.2.1 Scenario 1: Time context</td>
<td>49</td>
</tr>
<tr>
<td>6.2.2 Scenario 2: Location context</td>
<td>50</td>
</tr>
<tr>
<td>6.2.3 Scenario 3: Event notification context</td>
<td>50</td>
</tr>
<tr>
<td>6.3 Survey</td>
<td>51</td>
</tr>
<tr>
<td>6.4 Interview</td>
<td>51</td>
</tr>
<tr>
<td>7 Results and Analysis</td>
<td>55</td>
</tr>
<tr>
<td>7.1 Pilot Study</td>
<td>55</td>
</tr>
<tr>
<td>7.2 Main Study</td>
<td>56</td>
</tr>
<tr>
<td>7.2.1 Defining the participants</td>
<td>56</td>
</tr>
<tr>
<td>7.2.2 Scenario 1: Time context</td>
<td>56</td>
</tr>
<tr>
<td>7.2.3 Scenario 2: Location context</td>
<td>57</td>
</tr>
<tr>
<td>7.2.4 Scenario 3: Event notification context</td>
<td>58</td>
</tr>
<tr>
<td>7.2.5 Natural</td>
<td>59</td>
</tr>
<tr>
<td>7.2.6 Scenarios</td>
<td>61</td>
</tr>
<tr>
<td>7.2.7 Usability</td>
<td>63</td>
</tr>
<tr>
<td>7.2.8 Privacy</td>
<td>64</td>
</tr>
<tr>
<td>7.2.9 Participant choice</td>
<td>65</td>
</tr>
<tr>
<td>8 Discussion</td>
<td>67</td>
</tr>
<tr>
<td>8.1 The impact of context</td>
<td>67</td>
</tr>
<tr>
<td>8.2 Privacy threat</td>
<td>70</td>
</tr>
<tr>
<td>8.3 Review of the design and implementation</td>
<td>71</td>
</tr>
<tr>
<td>8.4 Review of the Methodology</td>
<td>72</td>
</tr>
<tr>
<td>9 Conclusion</td>
<td>75</td>
</tr>
<tr>
<td>10 Future Work</td>
<td>77</td>
</tr>
<tr>
<td>Bibliography</td>
<td>79</td>
</tr>
<tr>
<td>Appendices</td>
<td></td>
</tr>
<tr>
<td>Appendix A Questionnaire</td>
<td>85</td>
</tr>
<tr>
<td>Appendix B Interview questions</td>
<td>89</td>
</tr>
<tr>
<td>Appendix C Data from survey</td>
<td>91</td>
</tr>
</tbody>
</table>
### List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Spoken dialogue system architecture.</td>
<td>9</td>
</tr>
<tr>
<td>2.2</td>
<td>Dialogue from Iron Man</td>
<td>10</td>
</tr>
<tr>
<td>2.3</td>
<td>Speech recognition process</td>
<td>11</td>
</tr>
<tr>
<td>2.4</td>
<td>Absolute semantic</td>
<td>12</td>
</tr>
<tr>
<td>2.5</td>
<td>Context-dependent</td>
<td>13</td>
</tr>
<tr>
<td>3.1</td>
<td>Scenarios</td>
<td>25</td>
</tr>
<tr>
<td>3.2</td>
<td>Coding process</td>
<td>28</td>
</tr>
<tr>
<td>3.3</td>
<td>The core variable and the three categories</td>
<td>29</td>
</tr>
<tr>
<td>5.1</td>
<td>Work system and application domain</td>
<td>39</td>
</tr>
<tr>
<td>5.2</td>
<td>Mail received</td>
<td>40</td>
</tr>
<tr>
<td>5.3</td>
<td>History queue</td>
<td>43</td>
</tr>
<tr>
<td>5.4</td>
<td>Context recognition</td>
<td>43</td>
</tr>
<tr>
<td>5.5</td>
<td>Spoken dialogue system architecture.</td>
<td>45</td>
</tr>
<tr>
<td>6.1</td>
<td>Setup</td>
<td>48</td>
</tr>
<tr>
<td>6.2</td>
<td>RFID reader with RFID tags</td>
<td>49</td>
</tr>
<tr>
<td>6.3</td>
<td>TV with IR sender and a remote outlet with a Telldus device</td>
<td>50</td>
</tr>
<tr>
<td>7.1</td>
<td>Results for scenario 1</td>
<td>57</td>
</tr>
<tr>
<td>7.2</td>
<td>Results for scenario 2</td>
<td>58</td>
</tr>
<tr>
<td>7.3</td>
<td>Results for scenario 3</td>
<td>59</td>
</tr>
<tr>
<td>7.4</td>
<td>Natural</td>
<td>59</td>
</tr>
<tr>
<td>7.5</td>
<td>Usability</td>
<td>63</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

This chapter provides an introduction to the subject of the thesis to help the readers understand the scope of this thesis. The first section in the chapter provides a summary on the subjects ubiquitous computing, ambient intelligence and spoken dialogue system and provides a foundation on the problem statement. Furthermore, the introduction also presents the hypothesis and the purpose, which has guided the research. The chapter concludes with an outline of the thesis.

1.1 Overview

For the past decades, computers has evolved from being the same size as a room to something that would fit in your pocket. Today we are in the personal computers era accordingly to Mark Weiser and John Seely Brown[1]. They defined three different eras which represent the major trends of computing. The first era is called the mainframe era, and in this era many people shared one computer. The second era is the personal computer era, where a person has their own personal computer. The third and last era is called the ubiquitous computing (UbiComp) era.

The term UbiComp was first introduced in 1991 by Mark Weiser. He visioned people and environments augmented with computational resources that provide information and services when and where desired [2, 3, 4], and where machines fit the human environment instead of forcing humans to enter the computer’s environment.[2]

Mark Weisers vision of UbiComp spawned many new similar visions. One of these visions was ambient intelligence (AmI)[5]. AmI can in short be described as an electronic system that is sensitive and responsive to the
CHAPTER 1. INTRODUCTION

Presence of people.

AmI has been presented as a new computing paradigm that will revolutionize the relationship between humans and computers. AmI enriches the environment by using sensors to analyze and understand the user's activities, preferences, intentions and behavior and have the ability to adapt to meet the users needs.[6]

The AmI technology will be integrated into everyday objects and the user's environment, such as a home environment. AmI[5] aims to embed the user's entire environment in order to improve productivity, creativity, and pleasure through enhanced user-system interaction. The interaction between people and their environment should be seamless, trustworthy and in a natural manner.

The home environment has become an ideal environment for UbiComp[7, p. 503 - 506], mainly because the it contains many devices to help with activities. These environments typically contain all sorts of devices which are used to assist with activities around the environment. The devices could be used for everything from reading to keeping in touch with family.

In the traditional paradigm of human-computer interaction(HCI) the user is forced to learn how to use computers and adapt in order to use them. This has created a restriction for some groups of people and lead to a so-called “digital divide”. In AmI, the traditional paradigm is replaced by a new one, human-centric, where the computer has to adapt for each individual user and learn how to interact with them. The human-centric design enables the technology to be easily adapted by the masses, including the community of the other side of the “digital divide”. [6]

Two requirements that are essential for AmI are context awareness and natural interaction. The most natural mode for humans to communicate with each other is spoken dialogue, which has been successfully applied to many aspects of human-computer interaction[8].

1.2 Problem statement

Spoken language understanding is a challenging topic because of the difficulties of natural language processing and analyzing the user's speech. One of the requirements of AmI is that it should provide natural interaction between the user and the computer and that the user is not forced to learn and adapt to the computer.[6]

One specific type of an AmI system is a spoken dialogue system, which establish a spoken dialogue between the user and the system. These system
could both recognize speech and generate speech as an output.

In order for the user to easily interact with the computer through spoken dialogue, the computer should learn and understand the user’s activities and intentions. This has led to one of the most enduring problems for spoken dialogue systems[9]: achieving natural interaction and a human-like dialogue between systems and users.

One method to measure this is by using the Gulf of Execution[10, p. 50], which is the difference between the user’s intentions and the allowable actions. By measuring the gulf we can get an understanding on how well the system allows users to interact with a system without using extra effort [10, 7, p. 86 - 87]. David Benyon[10, 7, p. 86 - 87] state the following regarding the Gulf of Execution: “A key issue for usability is that very often the technology gets in the way of people and the activities they want to do.” and continues with “Very often when using an interactive system we are conscious of the technology; we have to stop to press the buttons; we are conscious of bridging the gulf.”. One of the goals of usability is to reduce the gap by removing steps that can cause distractions to users.

In order to achieve usability it is required to take a human-centred approach and try to achieve a balance between the four factors of human-centred interactive systems design: People, Activities, Context and Technologies (PACT)[7, p. 85].

Context needs to be analysed because it always exists in activities. Context can be seen as the feature that connects some activities together[7, p. 36].

By presenting context to voice commands, it could make the gulf of execution smaller and make interaction between users and computers more natural and comprehensible.

Three different terms are important to define before conducting this study:

| Natural: | In context of interaction, the term natural refers to when human-computer interaction resembles human-human interaction. The more human-like the interaction is between a system and a user, the more natural it is. |
| Usability: | The term usability is the ease of use and learnability of a system. When a system is easy to use and easy to learn, it has a high usability. If the system is hard to use and hard to learn, then it has a low usability. |
| Privacy: | Privacy can be defined as the right to be alone and the |
ability for people to seclude themselves or information about themselves. In computer science, privacy is often used when referring to preventing personal data from being collected and misused by others.[11, p. 5]

The following questions has guided this research:

- Could context-dependent voice commands in a spoken dialogue system increase the usability and make the dialogue more natural?
- Could context-dependent voice commands cause a privacy risk?

### 1.3 Hypothesis

The hypothesis for this research is that context-dependent voice commands can increase the usability for a spoken dialogue system and be more intuitive than absolute semantic commands for the user. By using context-dependent commands in the communication between the user and the system, it could feel more natural rather than using absolute semantic commands.

### 1.4 Purpose

In this thesis, I investigate what effects context-dependent voice commands have on the interaction between users and computers when using a spoken dialogue system. Furthermore, I investigate if there exist a potential privacy violation risk from the users’ perspective when using context-dependent commands with a spoken dialogue system.

### 1.5 Impact

This study should be of interests for spoken dialogue systems and user experience researchers. The result from this study should also be of an interest for the HCI-community, due to the study’s focus on human-computer interaction.

### 1.6 Delimitations

This research is mainly focusing on people using spoken dialogue systems in the home environment. Further studies could be performed to investigate
the implications context-dependent commands could have on people with various disabilities and how it could improve their condition.

The study will not go into algorithm detail on how the user’s voice could be interpreted, but rather the focus is on how to create context-dependent commands and how the user could execute them.

1.7 Outline

Chapter 1: Provides an introduction to the research field, which ultimately leads to the research question and the goal of this thesis.

Chapter 2: The initial literature review is presented and stated in Chapter 2 “Background”.

Chapter 3: In Chapter 3 “Methodology”, I present the the methods that were used to collect and analyse the data.

Chapter 4: In Chapter 4 “Related work”, I discuss and examine related work and studies for this research.

Chapter 5: The design of the spoken dialogue system are presented and discussed in Chapter 5 “Design”.

Chapter 6: The implementation of the spoken dialogue system are presented in chapter 6 “Implementation”.

Chapter 7: The collected results are presented and analysed in Chapter 7 “Result and Analysis”.

Chapter 8: The results of the study and the design decisions are discussed in Chapter 8: “Discussion”.

Chapter 9: A conclusion is reached and discussed in Chapter 9 “Conclusion”.

Chapter 10: Future works are presented in Chapter 10 “Future works”
Chapter 2

Background

The results of the literature research are presented in this chapter. The chapter also aims to introduce the research fields of spoken dialogue systems, ubiquitous computing, ambient intelligent, and human-computer interaction.

2.1 Spoken dialogue systems

A spoken dialogue system is an interactive system that has the capability to use speech as both input and an output. These systems use speech recognition to understand the user’s input and use generated speech as an output. Spoken dialogue systems differ from other similar types of systems, such as simple speech understanding systems[12], mainly because it can take context into account when analyzes the user’s input. The context is created when the user and system exchange information and interacting with each other. The interaction between a spoken dialogue system and a user is typically consisting of many turns of exchanges, but it can also be minimal and consist of only one exchange.

There have been three generations of spoken dialogue systems: Informational, Transactional and Problem solving. In the first generation, Informational, the systems retrieves information for the user. In the second generation, Transactional, the systems can assist the user in transactions. In the third and last generation, Problem solving, the systems provides assistance and support to the user in solving problems.[8]

Today, there exist several types interactive speech systems that allow the user to interact with their system, such as CHAT[13], TALK[8] and COMPANIONS[14] (see chapter 4). There are also research laboratories
across the world trying to develop more advanced systems that have the
capability to perform difficult tasks with a more conversational interface.

There exist many different types of spoken dialogue systems. The
following list outlines some of characteristic of each type of system:

Voice control: Enables the user to control the environment using speech. For example, a driver can control various devices when driving a car.

Call routing: Classifying a customer’s call and routing to the correct destination.

Voice search: Searching for information by using spoken query. The system will create a response to the spoken query, with information usually found in a database.

Question answering: These systems will provide answers to questions that were asked by using natural language.

Spoken dialogue: These systems help the user to perform well-defined tasks. The tasks are defined in advanced and system is designed to perform these tasks.

Even though many of these systems use different technologies, all of them have the same high-level architecture (figure 2.1). A user gives the system a spoken command which will be recorded and analysed by the Automatic Speech Recognition (ASR). The ASR will also transform the speech into text. The spoken language understanding (SLU) component will analyse the text and produce an output of formal representation of its semantics. In the traditional approach there exist two stages to determine the meaning of the text: syntactic analysis and semantic analysis. In the first stage, syntactic analysis, the SLU determine the constituent structure of the text. In the second stage, semantic analysis, the SLU determine the meaning of the constituents.

The dialog manager (DM) will process the representation and decide an output to the user. The external knowledge component will help the DM with processing the input in relation to the task at hand and its context, such as history. The Response Generator(RG) will compose a message based on the result of the processed representation. The message will be translated into speech by the text-to-speech synthesis component and delivered to the user.
The heart of this architecture is the dialogue manager[8] . The dialogue manager interpret the input, interacts with external knowledge sources and generates an output message to the user. Indeed, the dialogue manager is the central component of spoken dialogue systems. The dialogue manager consist of two tasks: Dialogue modeling and Dialogue control. The first task, the Dialogue modeling, is to keep track on dialogue state and provide the information used in dialogue control. The second task, Dialogue control, is to decide what will be the system’s next action based on the context of the current dialogue state. These decisions can be pre-scripted and based on the user’s confidence level. The system has a confidence threshold and if the user’s input is higher than the threshold then the system can interpret the input and decide the next action. The system will ask the user to repeat the input if the previous input could not be interpreted, i.e if the input’s confidence level was too low.

The confidence level could create a dilemma. If the system’s threshold is too low the system could interpret an input incorrectly and perform the wrong action. On the other hand, if the system’s threshold is too high there is a risk that the system will never allow any input. The system should have a balanced threshold that is neither too high, nor too low.

\(^1\)An adaptation of figure 9.1 from [8]
Jarvis: Yes. Shall I render using proposed specifications?
Tony Stark: Thrill me.
Jarvis: The render is complete.
Tony Stark: A little ostentatious, don’t you think?
Jarvis: What was I thinking? You’re usually so discreet.
Tony Stark: Tell you what. Throw a little hotrod red in there.
Jarvis: Yes, that should help you keep a low profile. The render is complete.
Tony Stark: Hey, I like it. Fabricate it. Paint it.
Jarvis: Commencing automated assembly. Estimated completion time is five hours.
Tony Stark: Don’t wait up for me, honey.

Figure 2.2: Dialogue from Iron Man

The spoken dialogue systems have appeared in several works of fiction, such as 2001 Space Odyssey and Iron Man (see figure 2.2). It is not unknown that fiction have been used as a source of inspiration for the development of technology. The English author and television personality, Karl Pilkington said the following words regarding the subject: “With all fiction comes the future.”

2.2 Speech recognition

Speech recognition is a critical part of human-centric interfaces and a core component that allow user’s to interact with spoken dialogue systems with their voice [15]. These kinds of systems used to be confined to research laboratories, but due to the significant progress made in recent years speech recognition is now being used in real-world application [15]. Systems that allow speech as an input have been available since 1990 [8]. In many of these systems the user can issue a command which the system will execute a required action to create a correct response to the user’s command [8].

A system’s ASR (see figure 2.3) has the objective to classify speech waveform into words, phrases or sentences. This process is typically made in two steps: Feature analysis of the speech signal and pattern classification. In the first step a sequence of feature vectors are produced. The product vectors contains data that characterizes speech utterances sequentially in time. In the second step, the sequence of feature vectors are compared against the machine’s knowledge of speech, such as acoustics, lexicon, syntax and semantics. This step is called the pattern classification and transform and transcribed data in the vectors to text [15].
2.2. SPEECH RECOGNITION

Even though speech input has not reached the same sophisticated level as speech output the technology has now reached levels of usability. The best ASR level are the ones that allow the user to train the system. A user can train a system by reading a specific text for the system. After 7 - 10 minutes of training an ARS’s recognition level can increase to 95% of accuracy.[7, p. 364 - 365]

2.2.1 Problems with speech recognition

Speech recognition has proved to be a challenging topic in natural interfaces. Even though there has been significant progress, there still exist some challenges and problems with speech recognition.

No guarantee: One of the main problem with speech recognition is that it cannot guarantee a correct interpretation. Noisy channels or noisy utterance makes it hard to interpret the original input correctly. This leads to the recognition system trying to make the best guess at what the original input was. [16]

Background noise: Noise in the background which affects the input signal.

Inter-speaker variability: The difference between how speakers speak and pronounce words. This problem is not as critical.

---

1 An adoption of figure 6.2 from [15]
Variability in speech signal: This problem occurs when there exist mismatch between the training and testing conditions. There are many reason for why mismatches between training and testing conditions occur. One reason could be background or ambient noise which affects the recorded speech. Another reason could be that the microphone used during testing uses a different frequency response than the microphone used for training. Some of the reasons could be because of the speaker. The speaker could also cause this mismatch. For instance, a person could pronounce a word differently depending on the state of health or the state of emotion that the speaker is in.[15]

2.3 Types of commands

2.3.1 Absolute semantic commands

An absolute semantic command is a command where the user has to explicit inform the system what the user want to achieve. These commands are very detailed and specific on what they want to achieve. The reason for this is because the system has to analyse the user’s commands solely on the information that could be retrieved from the command. If the commands are not detailed enough there’s a risk of misinterpretation (see figure 2.4).

![Figure 2.4: Absolute semantic](image-url)
2.4 Ambient Intelligence

The concept of ambient intelligence was first introduced by the company Philips in 1999. The term was used to represent their vision of futuristic technology[7, p. 490]. AmI aims to embed the user’s entire environment in order to improve productivity, creativity, and pleasure through enhanced user-system interaction. The technology will be integrated into everyday objects and the user’s environment. The word ambience itself refers to the need for large-scale embedding of technology. The word intelligence refers to social interaction between the user and the environment. The environment should have the ability to recognize people, personalize to their individual preferences, act upon the user’s behalf. The AmI vision places the human in the center and the human’s needs as the key element. The interaction between people and their environment should be seamless, trustworthy and in a natural manner [5]. People will be able to live and work in an intelligent environment which understands, recognize and respond to them [17].

The AmI original formulation contained four system elements [5]:

- Context-aware: The environment can sense and collect data. The data can be identified and categorized based on the context.
Personalized: The environment can adapt and be personalized in order to achieve the needs of the user.

Adaptive: The environment can adapt and change to match the user’s needs.

Anticipatory: The environment can respond to the user’s behaviour without conscious mediation.

AmI also includes the technologies of Intelligent User Interfaces (IUI), which is based on human-computer interaction research. This research aims to make interactions with computers more efficient, intuitive and secure by using more advanced interfaces, rather than the traditional interfaces like keyboard and mouse. The two key features for IUI are profiling and context awareness [17]:

Profiling: The system has the ability to make the environment personalized and adapted to different users.

Context awareness: The system has the ability to adapt to the situation.

In order to function, both of these features depend on sensors to record both the user and the environment. For example, a sensor could identify different users by their voice or face and detect moods and emotions by analyzing the user’s voice and body language. AmI technology also provides output and just like the input it is multimodal. An output could be everything from speech, such as spoken dialogue systems, to graphics and in various combinations. [17]

In Emile Aarts and Boris de Ruyter’s article “New Research perspectives on Ambient Intelligence” [5] they introduce three elements of social intelligence into the AmI environment. One of these elements were Socialized: The user should interact with the environment in manners which apply social rules of communication. By introducing this element, and the two other elements, they hope that AmI technology will meet the increased expectations of true intelligence. In order to reach true intelligence, the AmI environment requires social intelligence and has the capability to engage in social conventions.[5]

2.5 Human-Computer interaction

The research field of human-computer interaction (HCI) is said to have been founded in 1982 in a conference in Gaithersburg[18, 19]. HCI is regarded as a
rather complex research field, mainly because it is an mixture of other fields, such as computer science, sociology, psychology and communication[19]. The field arose when computers went from the mainframe era to the personal computer era[1]. At that time computers were marketed as a product to home-users and it also became an important tool to help many people with their jobs. Many of these peoples had limited training with computers and almost no technical experience. This created a “digital divide”[6]. It became important to make the interaction between all sort of people and computers easy and natural. In order to solve this problem, the field of human-computer interaction was created[19]. The HCI field has really evolved since its creation. Today is not only about how to improve peoples productivity, but also how to shape the everyday life and how we communicate with each other.[18].

2.5.1 Usability

Jeffrey Rubin and Dana Chisnell[20, p. 4] describe usability in short as “absence of frustration”. They later define it as “when a product or service is truly usable, the user can do what he or she wants to do the way he or she expects to be able to do it, without hindrance, hesitation, or questions.”. According the Rubin and Chisnell, usability contains six attributes, which all make a product or service usable: useful, efficient, effective, satisfying, learnable, and accessible.

Usefulness: “The degree to which a product enables a user to achieve his or her goals”.

Efficiency: “The quickness with which the user’s goal can be accomplished accurately and completely”

Effectiveness: “The extent to which the product behaves in the way the users expect it to and the ease with which users can use it to do what they intend”

Learnability: “The user’s ability to operate the system to some defined level of competence after some predetermined amount and period of time”

Satisfaction: “The user’s perception, feelings, and opinions of the product”

Accessibility: “Accessibility is about having access to the product needed to accomplish a goal”
An AmI system and a spoken dialogue system should be natural and easy to use. The system's usability value should be high in order to achieve this. When a system has a high degree of usability it is efficient, effective, easy, safe and should have high utility that allow people to do want they want to achieve.[7, p. 84 - 85]

2.5.2 Human-centric design

Being human-centred for an interactive system is about placing peoples’ needs first. A system should be designed to support people and for people to enjoy [7]. The human-centric design could also have many different interpretation depending on the application or technology context. In the book “Human-Centric Interfaces for Ambient Intelligence”[6] the authors, Hamid Aghajan, Ramón López-Cózar Delgado and Juan Carlos Augusto, state that even though human-centric design could be interpreted in many ways, it truly refers to a new paradigm where technology serves the user in whatever form. They state four parts to define the human-centric design paradigm: privacy management, ease of use, unobtrusive design and customization.

Privacy management: The human-centric design takes the user’s privacy into consideration in order to protect the user and the user’s integrity. An example of this is in vision-based reasoning, where “smart cameras” observe the environment and the users and abstract information. The processed pictures are deleted after the information has been collected.

Ease of use: Human-centric design systems are intuitive and easy to use. The user is not forced to learn the systems and therefore could be adapted by more people. This could result in technology being adopted by people on the other side of the “digital divide” and therefore reaching a larger mass.

Unobtrusive design: By using sensors a user’s productivity could increase in smart environments. By placing the user in a smart environment, the sensors in the environment could observe and interpret user-based events and attributes.

Customization: A human-centric system should be customizable to better serve the user and provide more accurate performance.
2.6. PRIVACY

In David Benyon’s book “Designing Interactive Systems” [7, p. 14] he states that being human centred for interactive systems is about four things:

- “Thinking about what people want to do rather than what the technology can do”
- “Designing new ways to connect people with people”
- “Involving people in the design process”
- “Designing for diversity”

2.5.3 Disability

Even though disability was not taken into account in this study, it is valuable to emphasize the impact AmI and spoken dialogue systems could have on people with disability.

One of the greatest advantages of smart homes and AmI environments are their ability to help people with their daily routine activities. Supportive homes with controllers and electric motors can help older people or people with certain disabilities[7, 506]. It is estimated that the percentage of the U.S population age 65 or older will rise from 13% to 19% from the year 2010 to 2030. U.S is not the only country that experience this trend. Other countries, such as Japan, estimates that the percentage of people of the age 65 or older will increase to 30% to the year 2033. Now many countries all around the world are asking the question how to pay for the care of the aging population. [21].

One method to reduce these costs is to embed peoples’ home with home sensors. This technology could allow people to stay healthy, but also be in their homes longer as they age. Home sensors could monitor peoples’ conditions in the same way clinics monitor conditions such as diabetes and congestive heart failure. Another method how sensors could function is to have them collect data in a context and then use it to infer information about everyday home behaviors. [21].

2.6 Privacy

Privacy itself can not be easily defined and yet it is considered to be a fundamental right to many people. Many countries see this as a right and laws have been created to protect it. The reason why it could be hard to create a general definition of privacy is because many people have their
CHAPTER 2. BACKGROUND

own definition of it[22]. Alan Westin said once: “no definition of privacy is possible, because privacy issues are fundamentally matters of values, interests and power”[23].

Throughout history privacy has had different general definitions. “The right to be let alone” was the definition to privacy by Samuel Warren and Louis Brandeis who wrote the influential paper “The Right to Privacy” in 1890[24]. Today, the word privacy tends to be used when referring to preventing personal data from being collected and misused by others.[11, p. 5]

Privacy has been said to be one of the biggest challenges for AmI[17]. There exist a concern on how AmI and sensor technologies will impact a user’s sense of personal privacy[25]. For the computers to be a part of the humans life and environment it has to monitor the user’s location, habits and even other personal information. This has to be done In order to achieve many of the AmI goals. Marc Langheinrich [24] describes four properties which makes ubiquitous computing different from other computer science domains:

- **Ubiquity:** One of the goals of ubiquitous computing is that it should be everywhere.
- **Invisibility:** Computers should be invisible for the user.
- **Sensing** Computers will use sensors to establish a state of the environment and the user.
- **Memory amplification:** Future applications of ubiquitous computing may record every action and movement of ourselves and our surroundings. This will allow the system and the user to search through our past.

AmI includes all these properties and with the IUI technology, two more properties are important in relation to privacy: Profiling and Connectedness. AmI can contain smart object, which can construct and use unique profiles of users. These object also have to be able to communicate with other devices.[17]

Emile Aarts and Boris de Ruyter asked the question: “What does it take for people to accept that their environment is monitoring their every move, waiting for the right moment to take over for the purpose of taking care of them?”[5]. This question can be defined as three basic questions about collecting data that should be answered:
2.7. ENVIRONMENTS

Data Collection: When will the data be collected and will it be invisible for the users.

Data Types: What type of data will be collected.

Data Access: Who will have access to the collected data.

When it comes to collecting data the question “who has access to it” always arises. A third party invades privacy if the party has access to personal information without the user’s knowledge or consent. Mark Weiser stated the following: “The problem, while often couched in terms of privacy, is really one of control. If the computational system is invisible as well as extensive, it becomes hard to know what is controlling what, what is connected to what, where information is flowing, how it is being used...and what are the consequences of any given action”[26].

2.7 Environments

Both UbiComp and AmI introduce new methods of interaction between computers and humans by sharing information throughout the environment, also called the information space. In these environments, many physical object will interact with each other and share information, but far from every object will be a computing object. An information space contains three types of objects: agents, devices and information artefacts. These three types of objects are introduced to provide a better understanding of the context in an information space.

Agent: An agent is a system that is actively trying to achieve a goal. An example of an agent can be people trying to perform their activity and reach their goal.

Device: A device is an object or component in the information space that is not concerned with information processing or can only receive, transform and transmit data but do not deal in information. An example of a device could be a piece of furniture or a button.

Information artefact: An information artefact is a system that can store information in a specific sequence, but also transform and transmit information. An example of an information artefact could be TV monitor.
In information spaces, people navigate through space and have to move from one information artefact to another. People could have access to both devices and to other agents in the same information space. The reason for defining these three types of objects is that one could easily create sketches displaying how the information is distributed through the components of a space. [7, p. 495 - 496]

The ideal environment for UbiComp is the home environment[7, p. 503 - 506]. Information and communication technologies have found their way into our homes ever since the start of the “information age". There are a number of general design principles for designing a futuristic home, also called a smart home [27]. One of these was that the technology should move to the background and that the interfaces should become transparent. Another principle was that the interaction with the home and its technology should be easy and natural to interact with.

2.8 Ubiquitous Computing

UbiComp is defined as the day when computing and communication technologies disappear into the fabric of the world[7, p. 489]. Using computers should be as refreshing as taking a walk in the woods [2]. In order to integrate information technology into a part of peoples’ lives, the computer has to move from being in the center to the background and become invisible. Weiser [2] stated the following: “The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.”. To reach the ubiquitous computing era the technology needs to achieve three requirements[2]:

- Cheap and low-power computers which includes convenient displays.
- Software for ubiquitous computing applications.
- Networks which ties all computers together.

The opposite to the ubiquitous computing vision would be virtual reality. The goal with virtual reality is to create a virtual world and place the user within it. The users will have to use special equipment in order to place themselves in the virtual world. Virtual reality focus on simulating the world rather than on invisible enhancing the world that already exists. ubiquitous computing resides in the human world and present no barrier to personal interaction. [2]
The strong opposition of these two notions resulted in the creation of the term “embodied virtuality”, which refers to the process of drawing computers out of their electronic shells. The goal for initially deploying hardware of embodied virtuality is to increase the amount of computers in the average room to hundreds of computers per room. These computers will be used like wires in the wall. The computers will become invisible to common awareness. Weiser stated the following about embodied virtuality: “By pushing computers into the background, embodied virtuality will make individuals more aware of the people on the other ends of their computer links”. Weiser proposed three basic devices for UbiComp:

- Tabs - Wearable pocket-size device
- Pads - Handheld page-size device
- Boards - Yard-size device

The smallest components of the embodied virtuality are tabs, which are interconnected to each other and will take on functions that no computers performs today. Tabs, pads, and boards are only the beginning. When these are starting to communicate and interacting with each other, the real power of the UbiComp concept emerges. For example, Weiser and his colleagues performed an experiment with embodied virtuality. In this experiment, doors open to the right badge wearer and telephone calls could be automatically forwarded to the user’s location. Weiser stated the following: “No revolution in artificial intelligence is needed, merely computers embedded in the everyday world”. [2]

This has lead to one of the fundamental challenges of UbiComp: How to push something to the background. Mark Weiser gives an example on how electric motors vanished in vehicles. By putting many small, cheap efficient electric motors into a single machine, each tool could get its own source of motive force. All these motors work together in order for the machine to function and most user are unaware of them[2].

Today, some parts of Mark Weiser’s vision has become true. We are using tabs, pads and boards which are interacting with each other. Technology and computers have become more integrated into our society. But there are still challenges which have to be solved before reaching Weiser’s vision. Gregory D. Abowd and Elizabeth D. Mynatt [4] outlined the four remaining challenges:

Natural interfaces: A more natural interface for communicating with computers, such as voice and gestures. The interaction between the
user and the computer will be less physical and more like human are interacting with each other.

Context-aware: The UbiComp applications should adapt the behaviour based on the content sensed from the physical world. More advanced sensors are needed in order to read complex content. A minimal necessary context are the “five W’s”: Who, What, Where, When and Why.

Capture: In order to introduce UbiComp for those who experience it late the UbiComp has to be flexible and provide universal access and strive to automate the capture of live experiences.

Everyday computing: Everyday computing present the challenge of time, where tasks do not have a clear starting or ending point.

Today the vision of UbiComp is the vision of the future, where the computers become invisible and embedded in the everyday objects. UbiComp marks the era where we truly reach the real potential of information technology.
Chapter 3

Methodology

In this chapter I present how to evaluate the users’ experience and how to collect appropriate data and how to analyse it. I also reach a conclusion on which data collection method, data analysing method and sampling method was appropriate to use in this study.

3.1 Literature review

The literature review was the initial phase for this research and consisted of reviews on spoken dialogue systems, AmI, HCI, Environments and also privacy. The findings of the literature review laid the foundation for this research and also directed the design process for both the spoken dialogue system and the test design.

3.2 Data collection

In the early days of HCI research the most measurements that were made were task-oriented [19]. Many of these tasks were based on human performance from human factors and psychology. These measurements are still considered to be the basic foundation for measuring interface usability. They can be used when the problem can be broken down into small specific tasks that can be measured in a quantitative way. Even though these measurement techniques are the foundation in HCI research, not all HCI problems can be solved by using this method. For instance, they are not appropriate when the task is about discretion and enjoyment. This kind of research question could not be answered by using quantitative methods. Instead a qualitative method could be used to answer this kind
of questions. Lazar, Feng and Hochheiser stated the following: “Direct feedback from interested individuals is fundamental to human-computer interaction (HCI) research”. Instead of going broad, we go deep and have a direct conversation with the concerned participants. By using direct conversation, new perspectives could be discovered that a survey might miss. A direct conversation usually takes two forms: interviews with an individual participant and a focus group involving many participants at the same time. [19]

A qualitative and quantitative method has been used as data collection methods. During the evaluation, the participant conducted a survey and answer questions about their experience. By using the survey the participant could answer every question immediately after experience the scenario. After the concrete scenario, the participants conducted an interview where the participant could extend his or hers answers in more detail. According to Jeffrey Rubin and Dana Chisnell[20, p. 19 - 20] this is one of the approaches for conduct a usability test and collect empirical data. This approach is used to confirm or refute specific hypotheses.

There exist three different types of interviews, structured, unstructured and semi-structured[28]. A structured interview is an interview where all the questions are decided before executing the interview. With a structured interview the margin becomes smaller that the interviews differ from each other, and therefore easier to compare the data from the participants. An unstructured interview is the opposite to a structured interview, where the questions do not have to be decided before the interview and instead of structured questions are terms used throughout the interview. A researcher may perhaps only ask one question during the interview, and let the participant answer the question freely. An unstructured interview reminds much of a normal conversation. A semi-structured interview is much like a structured interview, but it is more flexible and allow new questions to be asked. Semi-structured interviews were used in this study because they were suited for this study. With semi-structured interviews the interview could be structured, but still allow the participants to more freely answer the questions. The questions were structured, but the questionnaire could ask followup questions if the questionnaire wanted the participants to expand the answer with more details.

One recommended technique for collecting data is the “thinking aloud” technique[20, p. 204 - 206], where the participants say out loud what they are thinking. This technique has proven to be appropriate to capture what the participants are thinking while testing a concept or a product. Even though this technique has been proven successful, it was rejected for this
3.3. METHODS AND MATERIAL

study because of the risk of the ASR analysing the wrong speech. Instead, the participants were given the option to write down their thought on the survey during the test session.

3.3 Methods and material

To evaluate the users’ experience the participants were placed in various scenarios[7]. Scenarios are useful to get an understanding and evaluation of the four stages of interactive system design (see figure 3.1). There are four different types of scenario [7]:

- **Stories**: Stories are the real-world experiences of people. People’s stories are presented in rich in context and also capture seemingly trivial details.

- **Conceptual scenarios**: Abstract descriptions in which some details have been stripped away.

- **Concrete scenarios**: Generated from abstract scenarios by adding specific design decisions and technologies.

- **Use cases**: Describes the interaction between people and devices. The case describes how the system is used and also describes what people do and what the system does.

![Figure 3.1: Scenarios](image)

\[^1\] An adaptation of figure 3.10 from [7]
3.3.1 Story

A short story was created in order to capture the user activities and context in which they occur. The following story has been used to evaluate the users’ experience.

Richard is sitting in front of his computer and working on a document. While Richard is writing the text a computerized voice in the background starts to speak: “Richard, you got a new mail”. Richards stops writing and think for awhile. When Richard has decided what to do with this new information Richard says: “Please, read it”. The computerized voice in the background starts to speak again: “The mail is from Peter and contains the following message: Hi Richard! I just want to inform you that the meeting from 1 pm, has been moved to 2 pm. Best regards Peter”. Richard could respond to this email, but choose not to. Richard instead start to write on his document again.

3.3.2 Conceptual scenario

The conceptual scenario is a user trying to use voice commands with context-dependent. The computer presented new information by using speech. The user gave a new voice command which was dependent on the previous given information. When the computer received the new command, it matched it with the previous given information and tried to interpret it in a correct fashion depending on the context.

3.3.3 Concrete scenario

In this study, the participants were placed in an environment with one computer. The user received information from the computer via a computerized voice. The user was then able to respond to this information with either context-dependent commands or absolute semantic commands. The user could afterwards receive feedback from the computer based, or other devices, on the result of the user’s previous command.

The following points had to be functional to evaluate the concrete scenario:

- Voice recognition
- Computerized speech
3.4. DATA ANALYSIS

- Absolute semantic command recognition
- Context-dependent command recognition
- Input from various information sources

The concrete scenario generated use cases, which were used to evaluate the users’ experience. Each scenario should simulate a realistic scenario. This will help the participant to focus on the scenarios. Jeffrey Rubin and Dana Chisnell[20, p. 182 - 184] state the following regarding the matter: “The closer that the scenarios represent reality, the more reliable the test results.”.

3.4 Data analysis

3.4.1 Survey structure

Even though the participants conducted a survey during the test scenario, it is not the main data for this study. The data collected from the interviews will serve as the main data for this study. The survey data will be presented and simple describe what has been collected. The answers from the conducted survey will be crossed analysed with the answer from the interviews.

3.4.2 Interview structure

To analyze the collected data two analysis methods have been considered: Grounded theory[28], defined by Glaser and Strauss, and Content analysis[19, p. 285 - 289], defined by Holsti.

Grounded theory is one of the most common methods in qualitative research. In Grounded Theory there is a close connection between the collection of data, analyzing data and the result. This make it possible to collect data until the research question is answered and a theory is created. Grounded theory is used to create a hypothesis based on observation or other kind of data. For this reason, Ground theory is not suited to used in this study because this study already have a hypothesis.

Content analysis is described as an analysis method that aims to generate new knowledge from an in-depth analysis. Content is divided into two categories: media content and audience content. The media content covers printed publications, broadcast programs or recordings. The audience content covers text (Notes from interviews or observation) and multimedia
(video- or audio-recording of interviews). Content analysis will be used as analysis method in this study because it provide an in-depth analysis and may provide information to answer the research question. One requirement for using Content analysing is that the study has to have a clear definition of the data that will be analysed. In this study the data that will be analysed is the participants impression on how natural and usable context-dependent commands are in comparison to absolute semantic commands.

In order to analyze the content of the collected text, “coding”[19, p. 289] was used. As described by Corbin and Strauss[29, p. 65], the process of coding “It involves interacting with data (analysis) using techniques such as asking questions about the data, making comparisons between data, and so on, and in doing so, deriving concepts to stand for those data, then developing those concepts in terms of their properties and dimensions”.

The coding process has two approaches to analysing the data: priori coding and emergent coding. Priori coding was used in this study because it more appropriate to use when there existed a sufficient amount for literature related to this study. By having related literature it becomes possible to establish coding categories. The other coding approach, emergent coding, is more suitable for when there exists a very limited literature from previous studies.

The priori coding[19, p. 289] process involves three stages: Identifying coding categories, Coding of text data and reliability check (see figure 3.2). In the first stage, categories are defined based on established frameworks theories. In this study, the theory is that context-dependent commands are more natural than absolute-semantic commands and that it increases the usability. In the second stage, coding of the text data is performed. The text data in this study will be the transcribed interviews. The coding involving categorizing the text data into different categories. The last stage is reliability check. In this stage, consistency is checked to ensure that the coding process were correctly performed.

Based on the literature study, there exist three categories that are of
interest for this study: Natural, Usability and Privacy (see figure 3.3).

![Diagram of core variable and categories]

Figure 3.3: The core variable and the three categories

Two different transcribing methods were considered for this study: Naturalized Transcription and Denaturalized Transcription[30]. Naturalized Transcription is not only transcribing all what the participants are saying but also include codes for when there are pauses in speech, not spoken activities and similar activities. Denaturalized Transcription is focusing in the informational content of the speech. The second transcribing method will be used in this study because this study is more concerned on the content of the speech, rather than how it was delivered.

Three guidelines from Gillhams book Research Interviewing will be used when transcribing the taped interviews. The first one is omitting interjections that do not add any meaning to the transcript. The second is the interview format: Interviewers speech is printed with bold style while the interviewees speech is printed with regular style. The third and last is appropriate punctuation, which means adding punctuation when transcribing making sure not to alter the meaning of the spoken words. [31]

### 3.5 Sampling method

Two sampling methods were considered for this study: theoretical sampling and convenience sampling[32]. Theoretical sampling is a sampling method where choosing data sources that are most suited to develop a theory and compare it to a previous research. Convenience sampling is a sampling method which excludes no data sources that are available at that time. Convenience sampling was the most suitable method and was therefore used as the sampling method in this study. The reason for choosing convenience
sampling is mainly because the study does not focus on a specific targeted group of people.

3.6 Orientation Script

The purpose of the orientation script[20, p. 155] is to brief the participant what will happen during the test section and about the testing setup, such as equipment. By explaining that it is the concept itself that is being tested and not the participants, will help to put them to ease and let them focus on the test session. The participants were briefed before they accepted to participate in the study, but also before they conducted the test session.

3.7 Ethical issues

All participants in the study was informed that participation were voluntary and that they could abort the experiment and end the interview at any time and there will be no consequences by doing so. The interviews was audio recorded and all the information collected in interview and surveys could be used in the report. No vital information about the participants was collected to ensure that the respondents could not be traced or identified. [33]

3.8 Technology

In order to create the concrete scenario and evaluate the users’ experience, a spoken dialogue system was required. The Wizard of Oz technique[34] was considered to be used to evaluate context-dependent commands and absolute semantic commands. The Wizard of Oz technique gives users the ability to test a real system, even before it is developed. Instead, users are interacting with a human, the wizard, that acts as a computer. The wizard will observe and respond in various ways to give the user the illusion that they are working with a real system. David Benyon[7] describes it as technology being replaced by humans.

Even though the Wizard of Oz technique is appropriate to use in this study, it was excluded. The reason for this was because the author already had experience with similar technique and could develop a fully functional spoken dialogue system. By using a real system, the user got a more real experience on what a spoken dialogue system is and how it is functioning with both context-dependent commands and absolute semantic commands.
3.8. TECHNOLOGY

The results will also be more reliable if the scenarios represent reality[20, p. 182 - 184].
Chapter 4

Related work

In this chapter, I provide previous work related to this study. The previous work can be containing similar technology or concepts mentioned in this study.

4.1 Google Glass

The Google Glass project\(^1\) was first introduced in 2012 and is being developed by Google X Lab\[^{35}\]. Google Glass is a pair of augmented reality head-mounted display (HMD) that has the ability to use Google services, such as search function, voice commands, take photos, location check-ins, maps, video conversation and much more. The glasses display information to the user through a display that is placed in front of the user’s right eye.

The glasses is connected to the internet and the user can interact with it through natural language voice commands. The user will be able to speak with the glasses and it will analyse the user’s command and act appropriately based on the command and the context. Today, these glasses only exist as a prototype, but they are expected to be available for consumers before the end of 2013\[^{36}\].

4.2 Siri

Siri\(^2\) is based on the fields of artificial intelligence and natural language processing and was developed by Apple Inc. It is described as an intelligent

\(^{1}\)http://www.google.com/glass/start/
\(^{2}\)http://www.apple.com/ios/siri/
personal assistant and a knowledge navigator. It allows users to interact and command the iOS platform throughout natural language user interface. Siri also provides dictation functionality instead of using the keyboard. It has three components: a conversational interface, personal context awareness and service delegation\cite{37}. By using statistics from the voice recognition and machine learning, Siri has the ability to adapt to the user’s individual preferences. The speech recognition technology\cite{38} was supplied by the company Nuance Communication Inc\footnote{http://www.nuance.com/}.

4.3 CHAT

The CHAT\cite{13} project is a spoken dialogue system that currently operates in the entertainment, navigation and web services domain. The CHAT technology provides robust speech recognition and spoken language understanding. Unlike most of the current spoken dialogue system systems that is tied to a specific domain, CHAT has the ability to change domain on the user’s command. This allows the user to change from controlling the music in the user’s environment, to ask for direction to a specific location.\cite{8}

4.4 TALK

TALK\cite{8} was a multimodal dialogue system and the main aim for the system was to make a dialogue system more adaptive, conversational and robust. In TALK, the theoretical basis for dialogue management is information state update (ISU), which provides a rich formalism for the representation of dialogue context. One of the main goals of the systems was reusability, the ability to reuse the representation of dialogue context for different languages, modalities, and application domains.\cite{8}

The systems SAMMIE\cite{39} and MIMUS\cite{40} are developed using TALK. SAMMIE is a in-car system that allows the driver to control the MP3 player with commands. MIMUS is a home automation system that allows people to interact with the environment using voice and mouse clicks. MIMUS has the ability to process unrestricted spoken dialogue which proved to be a great advantage because users did not have to learn sets of commands.\cite{8}
4.5 COMPANIONS

COMPANIONS[14] is a system that aims to assist senior citizens in daily activities. The main technologies of the system are speech recognition, natural language processing and emotion processing. The system can carry out everyday tasks and easily access information by using devices, such as PCs and handhelds. To demonstrate COMPANIONS, two companions were developed: Health and fitness companion and Senior companion. The health and fitness companion help users to support a healthy lifestyle by monitoring information about the user’s eating habits and fitness activities. The purpose of the Senior companion was to help seniors with everyday tasks and provide access information easily. The companion could even chat with the users and entertain them.[8]
Chapter 5

Design and implementation

This chapter presents and discusses the design and implementation phases of the spoken dialogue system used during this research. It also explains the difference between absolute semantic commands and context-dependent commands and how these two types of commands were implemented in the system. The design phase lay the foundation to the implementation phase, where I present the different modules that have been implemented and used when performing the experiment and collecting the data. This chapter also presents the solution on how the speech recognition and other functions were implemented.

5.1 Design goal

In order to investigate and collect data for this research, a spoken dialogue system was required. The system had to provide the basic functionality of a spoken dialogue system, i.e support both speech input and output. Furthermore, the system also required to support both absolute semantic commands and context-dependency commands. The system had to be capable to be configurable and function as a central component and listen for events from other sources, such as mail. The users should be able the activate functionality by using voice commands, but also act upon incoming events from external sources. In order for the user to refer back to the context, the context and the information has to be stored and organized by the system. Otherwise, there would be no context to use when interpreting incoming voice commands from the user.

Even though voice recognition software has become more available, no spoken dialogue system was found that match these requirements.
Because of this, the author choose to develop a system that fulfilled these requirements.

5.2 Spoken dialogue system

The developed spoken dialogue system was mostly developed in Java by the author and is called “The interactive system”. Java is an object-oriented programming language which was developed in the early 1990s by Sun Microsystems[41]. The system enables the user to create their own environment and configure it however they see fit. The system contains the following components: SLU, DM and RG. This means that it will process and analyse speech, managing dialogue modeling and dialogue control, and also generate a result based on the speech input. The system was also capable of interacting with external sources. The system had a human-centric design and was designed to be used in a home environment. Therefore, the system was capable of handling both agents and information artefact. Examples of information artefacts in the home environment could be lights and a television.

The system was partly designed as an ambient intelligence system and embedded technology into the user’s environment. By doing this, the user could receive feedback from an information artefacts when a voice command was interpreted successfully.

The system was more adapted to Apple’s operating system, Mac OS X, but because the system was developed in Java it has the ability to be adapted to a Linux or Microsoft operating system. The main reason for choosing to develop the system in Mac OS X, is because the system was communicating with external software functioning in Mac OS X only.

The system had three main components: Input, Subsystem, and Output. All of these components inherited from the class Module. The user could choose which modules should be used in an environment and which modules should be able to interact with each other. The input module allowed users to communicate with the system and activate functions in the subsystems. Each subsystem represented a different area of functionality. A subsystem could interact and control other applications or objects in the environment. When the requested function had been executed, the output component would publish the results. All these components were developed to run on a specific operating system. This means that some of the components would only work with a certain operating system. There also existed components that were not tied to a specific operating system.
There are two types of components which are user defined: Command and Connection. A connection functions as a bridge between inputs, commands, and outputs. By using connections a user is able to define which inputs, commands, and outputs should be able to interact with each other. The command component is a user defined command which upon being triggered can activate multiple functions in different activated subsystems. The user can give a command keywords and a direct command word to execute the command. When executing a command the system will check with keywords and the direct command. By defining the connection, the user can connect these component and can configure which components should be able to affect other components.

5.3 Task design and analysis

A task can be seen as a specific view of interactive system design that leads to specific techniques. The task concept can be viewed as people interacting with technology. Together they are sometimes called a “work system” and are trying to achieve changes in an application domain (see figure 5.1). An application domain is an abstraction of the real world that could be representing an external entity, such as databases or web pages. When analysing tasks one should distinguish between three key concepts: goals, tasks and actions. In short they can be identified as follows: “A task is a goal together with some ordered set of actions”.[7]

![Figure 5.1: Work system and application domain](image)
The goal is the state of the application domain that the work system want to achieve. In order to achieve this goal, the work system has to execute a structured set of activities, i.e. a task. A task normally consist of subtasks and each subtask is a detailed task. The tasks are broken down into subtasks until the task is defined as an action. Actions are simple tasks and can be executed without having a problem solving associated with it.\[7\]

In a similar fashion, the design on a user’s task and command in a spoken dialogue system can be divided into subcommands. When a command has been executed its subcommands are also allowed to be executed. Dividing a command into subcommands allowed the user to execute specific subcommands which have a relation with the previously executed command.

An example of this is an email update. When the system receives an email update it will execute a command which will inform the user about the latest received email. The user can then react upon this update and the system will match the user’s reaction to a set of subcommands. In this example, the user could either say “Read it” or “Discard it” (see figure 5.2). These actions can never be executed by the user, unless the previous command has been executed. The user could choose to execute the same subcommand without having the previously command be executed, but then the user has to use absolute semantic commands. In this example the user would have to something like: “Read the latest email received”.

\[1\] An adaptation of figure 11.1 from \[7\]
5.4 Absolute semantic commands

The system has the functionality to recognize absolute semantic commands spoken by the user. Instead of using context when interacting with the system, the user can choose to use absolute semantic commands. As presented before (see page 12), an absolute semantic command differ from context-dependent commands because of the lack of context. Absolute semantic commands have no knowledge of any prior interaction between the user or the system.

Algorithm 1: Execute absolute semantic commands

<table>
<thead>
<tr>
<th>input : input component, message from input</th>
<th>output: Execution results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 for connection ∈ connections do</td>
<td></td>
</tr>
<tr>
<td>2    if (connection ∉ input) then</td>
<td></td>
</tr>
<tr>
<td>3    continue;</td>
<td></td>
</tr>
<tr>
<td>4 end</td>
<td></td>
</tr>
<tr>
<td>5 for command ∈ commands do</td>
<td></td>
</tr>
<tr>
<td>6    if (ConfidenceLevel(command, message) &gt;</td>
<td></td>
</tr>
<tr>
<td>ACCEPTANCE_BAR ) then</td>
<td></td>
</tr>
<tr>
<td>7      result ← Execute(command, message)</td>
<td></td>
</tr>
<tr>
<td>8      WriteToOutputs(connection, result);</td>
<td></td>
</tr>
<tr>
<td>9      activeCommands ← activeCommands ∪ {command};</td>
<td></td>
</tr>
<tr>
<td>10     return;</td>
<td></td>
</tr>
<tr>
<td>11 end</td>
<td></td>
</tr>
<tr>
<td>12 end</td>
<td></td>
</tr>
<tr>
<td>13 end</td>
<td></td>
</tr>
</tbody>
</table>

Every command has the ability to be triggered by using keywords (see algorithm 1). The system will match all received inputs with keywords. A command will only be executed when the confidence level is high enough. The confidence level can be configured by the administrator of the system. For example, the user might want that all keywords exist in the received input, i.e. a confidence level of 100%. If a command contains the keywords “read” and “it”, then the input has to contain both of these words in order to have a high enough confidence level to execute the command.
5.5 Context-dependent commands

A context-dependent command shares many similar features as absolute semantic commands, but as the ability to use context to analyse and interpret the command. This allows the user to be less specific when interacting with a spoken dialogue system, because it is relying on context to fill the gap (see algorithm 1).

<table>
<thead>
<tr>
<th>Algorithm 2: Execute context-dependent commands</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>input</strong>: input component, message from input</td>
</tr>
<tr>
<td><strong>output</strong>: Execution results</td>
</tr>
<tr>
<td>1 for connection ∈ connections do</td>
</tr>
<tr>
<td>2 if (connection ∉ input) then</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5 for command ∈ commands do</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>14</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>16 end</td>
</tr>
<tr>
<td>17 ExecuteAbsoluteSemantic(input, message);</td>
</tr>
</tbody>
</table>

When a context-dependent command is executed in application domain, it will examine the command and which state the application domain is currently in. The application domain will search through the history of executed commands and try to match the new user input to a subcommand (see algorithm 2). The system will try to execute a regular command (see algorithm 1) if no subcommand had a high enough confidence level. When a command has been executed, it will be placed in a history queue. This data structure operates supports the FIFO order, “first-in-first-out”. [42].

The executed command will expire after a certain time, due to being out of date, and be removed from the queue.

![History queue](image)

Figure 5.3: History queue

When the application domain receives a new input it will iterate through the history queue and examine all the commands and their subcommands (see figure 5.4). It will start to examine the latest executed commands. The domain administrator can configure the size of the queue and how long time a command is placed in the queue. If the application domain finds a subcommand that match the input it will be executed and placed in the history queue. The input will be examined as an absolute semantic command if no matching subcommand is found in the history queue.

![Context recognition](image)

Figure 5.4: Context recognition

### 5.6 Implementation

The spoken dialogue system is divided into three parts: Automatic Speech Recognition, Text-to-Speech Synthesis and the rest, which contains spoken
language understanding, dialogue manager and response generation (see figure 5.5).

The software Dragon dictate\(^1\) developed by Nuance Communication Inc was used as the ASR component. Dragon provides functionality to recognize speech from the user and translate it to clear text. The software also provides a set of preconfigured commands to interact with the computer, such as control the web browser. Later in this chapter we will discuss how Dragon dictate allows us to communicate with our system and how they interact with each other.

The second part of the solution is the Text-to-speech synthesiser. This part allows the system the interact with the user throughout generated speech. The operating system, Mac OS X, has the functionality to generate speech and play it to the user. This was one of the reasons why Mac OS X was chosen to be the platform for the system. The Interactive system can use AppleScript to interact with the operating system and certain applications. AppleScript\(^2\) is a scripting language which is integrated with the operating system and provides the functionality to command scriptable application and many part of the Mac OS X. By using the Terminal command “say” and a given output, the operating system can generate the given output as speech.

The last part is the Interactive system, which was presented before in this chapter (see page 38). The interactive system acts as a core and handles input from the automatic speech recognition software, analyzes the input together with the stored context, interacts with external sources and generates a response.

In the remaining of this chapter I will present the technology that was used during the test study. The technology and its devices was used to simulate a home environment.

5.6.1 Speech Recognition

As presented before in this chapter (see page 44), the speech recognition was accomplished by the software Dragon Dictate. A user defined command was created in Dragon Dictate which was triggered when the user said the word “Computer”. When the user said this word, the software started to transcribe the user’s speech. In order to end the transcribing phase and

---

\(^1\)http://www.nuance.com/

\(^3\)An adoption of figure 9.1 from [8]
commit the transcribed speech, the user said the word “Execute”. Dragon dictate uses AppleScript and the service Netcat to send an UDP message containing the transcribed speech to the Interactive system. The interactive system acts as a UDP server and constantly waits for input. Upon received UDP message, the system starts to analyse accordingly to either algorithm 1 or 2. The system will execute every command in the same system connection as the UDP server if there is an accurate match. Otherwise, the system will be unaffected by the UDP message.

There is a second method to activate Dragon Dictate instead of saying the word “Computer”. The system may act upon external inputs, such as received mail or time alarm, which can activate Dragon Dictate transcribing phase. The user could predefine how long time the software will transcribe. Regarding if the time or the user end the transcribing phase, the following event will act the same as stated earlier.

5.6.2 RFID input

The Radio Frequency Identification (RFID) input was used to activate the scenarios which the participants conducted. The RFID hardware and
software was developed by the company Philgets Inc\textsuperscript{4} and integrated with the Interactive system. In the Interactive system, the user can register several RFID tags and link them to user defined commands. When a RFID tag is scanned by the RFID reader, it activated the linked user command. In this study, every scenario was linked to a unique RFID tag.

### 5.6.3 TV subsystem

The TV subsystem allows the user to control TV, or other devices, with infrared light (IR). The external hardware and the Java library was developed by the company Philgets Inc\textsuperscript{4}. By implementing the external library from Phidgets into the Interactive system, the user could send IR signals and control devices, such as the TV.

### 5.6.4 Lights subsystem

The light subsystem allows the user to control remote outlet, e.g turn on and off lights. The used hardware, which enables us to perform these task, is called TellStick and is created by the company Telldus Technologies\textsuperscript{5}. TellStick is using radio waves to operate the remote outlets. The company provides an API for controlling the TellStick through many different programming language. There were some problems when developing a Java interface for controlling the device, which led to a different solution. Instead of controlling the device from Java, I created a Python interface, which acted as a server that could receive UDP messages and control the TellStick. Python is a high-level programming language created in the early 1990s\textsuperscript{[43]}. The interactive system sends a plain text message containing information about which remote outlet should be turned on or off.

### 5.6.5 Spoken output

The spoken output was provided by using the integrated functionality, in Mac OS X, to generate spoken dialogue. The spoken output takes a given message and uses AppleScript to generates speech and play it for the user.

\textsuperscript{4}http://www.phidgets.com/
\textsuperscript{5}http://www.telldus.se/
Chapter 6

Test design

In the following chapter, the test design is described, covering the room setup and the tasks that the participants have conducted in the experiment. The tasks are divided into scenarios which are described in detailed how the participants could perform them with and without context. The chapter also presents the survey and interview questions.

6.1 Setup

A room in the Department of Computer and Systems Sciences (DSV)\textsuperscript{1} at Stockholm University (SU) was chosen as location to conduct the experiment and interviews. The reason for choosing this location is because the author, was at the moment of conducting the experiment, employed at DSV and had access to these locations.

The size of the room was 10m\textsuperscript{2} and has one window (see figure 6.1). The room contained one laptop (MacBook Pro), one desktop lamp, three chairs and two desks. The participants were placed in front of a desk with the laptop. The computer displayed the results of the speech recognition, before it was analysed by the interactive system. The computer was also used as a microphone and the participants did not directly interact with the computer. Each participant also conducted a survey during the test, so they had all the material need to conduct the survey available.

When participants arrived to the location and were ready to conduct the experiment, they were informed about the study and the terminology according to the predefined orientation script. One important thing the

\textsuperscript{1}www.dsv.su.se
6.2 Scenarios

Scenarios are tasks performed by the participants that represent actual work which a user could perform with a spoken dialogue system[20]. Each scenario explained in the section was performed with both absolute semantic commands and context-dependent commands. Each scenario let the user interact with either external devices or software on the computer. The scenarios were ranked accordingly to the level of differences between absolute semantic commands and context-dependent commands. The scenarios were executed by using unique RFID tags (see figure 6.2).
In the first scenario, the absolute semantic commands and context-dependent commands were similar. On the second scenario they are starting to differ more and take more context into consideration. The third and last scenario allowed the user to fully use the advantages of context. The reason for this design was because the participants, in the beginning of the test study, were supposed to question if context really is necessarily. At the end of the test study the participants should have a clear perspective on how context can be used and when it could be used.

### 6.2.1 Scenario 1: Time context

The first scenario in this experiment simulated how the user could interact with a television. The participants were informed by a computerized generated voice that their favorite program was about to start. In this scenario, both the time and the dialogue is the context. When using absolute semantic command, the participant had to explicit explain to the computer what the participant wanted to achieve. The participant could say something similar to: “Start the television”. When the participant was using context-dependent commands, the participant only had to say the words: “Start it” or “Show it”, and the computer would understand that the participant was referring to the television (see figure 6.3).
6.2.2 Scenario 2: Location context

The second scenario in this experiment simulated how lights could be turned on and off with context and without context (see figure 6.3). This scenario demonstrated both context in location and in the dialogue. The goal of this scenario was to turn on and off the lights with both context-dependent voice commands and absolute semantic commands.

When the scenario was using context, the user could say: “Turn on the lights” to turn on the lights in the room. When conducted this experiment there was only one lamp in the room and therefore there are no reason to keep track of the user’s position. This scenario only provides the illusion of location context. When the scenario was not using context, the user had to be more specific on which lamp he or she wanted to turn on or off. For example, without context the user had to say the following to turn on the lights in the office: “Turn on the lights in the office”.

6.2.3 Scenario 3: Event notification context

The third and last scenario in this experiment simulated an interaction between the user and the computer regarding an email event. This scenario demonstrated how the user’s commands are more a part of the dialogue with context, rather than not having context at all. The computer would inform the user that a new message had been received. The user could react upon this new event and command the computer to read it or not. When the scenario was using context, the user could simply say the words “Read it” and the computer would understand that it was the mail the user was referring to. If the scenario is not using context, then the user had to provide an absolute semantic command: “Read the latest received mail”.

Figure 6.3: A: Samsung TV, B: IR sender, C: remote outlet, D: Telldus device
The computer read the received email when the user had correctly reacted upon the received email event. If the scenario is using context, the computer would ask the user if he or she would like to reply to the email. The user could then answer with a yes to start composing a new mail to the sender. If the scenario was not using context, the user had to tell the computer to reply to the latest mail received. In both scenarios the user had to say “Send” when the user was done composing the email and wanted to send it.

6.3 Survey

During the three scenarios, the participants conducted a survey (see appendix A). The purpose of the survey was mainly for the participants to write down their feelings and reaction of each scenario. The answers from the survey helped the participants to recall and remember their feeling from each scenario when they conducted the interview. The answers from the survey are also presented as figures in the result chapter.

The first questions are asked in order to define the participants. Two questions was used to define the participants: Their age and the computer knowledge level. After each scenario, the participants answered questions regarding how natural the conducted scenario was and how usable it was.

6.4 Interview

In order to successfully answer the research questions, the interview contained questions regarding how natural the interaction was, if the participants felt that context could improve or impair the usability, and if the user felt that this could harm their privacy (see appendix B). Therefore eight questions were chosen to evaluate the user’s experience. In the following, eight questions are presented and the reasoned for their usage is stated.

- When did you experience that the dialogue became more natural: Using context-dependent commands or absolute semantic commands. This question was the first question the participants were asked in the interview. The participants had a chance to decide which of the two alternatives, absolute semantic commands or context-dependent commands, made the dialogue more natural. The participants that did not elaborate why they choose one over the other, were asked to elaborate further and explain in detail.
• When did you experience that the usability was at its highest? The second question in the interview examined the usability of context-dependent commands or absolute semantic commands. Moreover, the question also strive to raise a question if usability is necessarily increased if the interaction between the human and the computer is more natural.

• Did you feel more monitored when having context in the dialogue, in contrast to using absolute semantic commands? The third question raises the third aspect in this study: Privacy. Because of the interaction between the user and the computer could be regarded as more humane, there could be a risk of the user feeling more monitored.

• Do you feel that the context in the dialogue could cause a threat to your privacy? The fourth question goes deeper on privacy issue and examine if there maybe exist a potential threat to the user’s privacy from the user perspective. The participant could answer if they experience that context causes a threat to the privacy, or if they think that context theoretically could be a potential threat.

• Which types of scenarios are suited for using context in dialogues and why? The fifth question is a rather free question and lets the participants use their imagination and find scenarios in the daily life where context could be suited. A scenario could be everything from the given scenarios from the experiment or from somewhere else. The participant will also get a chance to elaborate for why a scenario is suited for a specific scenario.

• Which types of scenarios are not suited for using context in dialogues and why? The sixth question examines where context is not suited and why it is not suited for the scenario. This question is a direct opposite to the previous question, which makes the participants try to define scenario or environments where context-dependent commands are not suited.

• Which types of commands did you prefer and why: Context-dependent commands or absolute semantic commands? In the seventh question of this test study the participants were asked which kind of command they preferred and why. Even though participants would say that context-dependent or absolute semantic commands
are more natural and useful, there could still exist a reason for the participant to prefer the other option.

- **Do you have any comments, remark on the experiment or the technology?** The last question in the interview provided an option for participant to give suggestion for further design decisions and development, as well as express their thoughts on the test session.
Chapter 7

Results and Analysis

In the following chapter, the qualitative and quantitative results from both a pilot study and the main study are presented. The results are derived from the introduced methodology from chapter 3. The findings and data categorized by either the scenarios or by the core variable of this study: Natural, Usability and Privacy.

7.1 Pilot Study

The aim of the pilot study was to test the study setup in order to reveal early findings. The environment was prepared as envisioned and equipment was installed and tested. Two participants, including the author, were placed in each scenario with the usage of both the context-dependent commands and absolute semantic commands. The participants were also briefed according to the orientation script.

The two participants were able to perform all scenarios and did not experience major interruptions. Both participants were able to interact with the environment as described in the test design.

One of the main worries for the experiment was the ASR. There could have been a risk of misinterpretation mainly because the participants did not have a lot of time to train the ASR. Luckily, there were no major misinterpretation during the pilot study.

The result of the pilot study led to some changes at the mail scenario and the interview questions. The response time for the participants were also increased. With increase response time, the participants would have a chance to repeat the missing keywords again in order to trigger the command. The interview questions in the pilot study were somewhat unclear.
and misleading according to the participants. Therefore, the questions were updated so that they were more understandable and did not mislead the participants to favor one option over the other.

7.2 Main Study

This section presents the findings of the main experiment. After a brief description of the participant selection, the observed and reported results of the study and their analysis are provided. The result from the survey is presented in appendix C.

7.2.1 Defining the participants

The main experiment was held during ten days with ten participants, one woman and nine men. Nine of ten participants were between the age 20 and 30, while one of the participant was over 40. All of the participants were non-native English speakers and had different origins. Eight of the participants chose to report their observations and opinions in Swedish, while the rest of them did it in English. All of the recruited participants were either undergraduate students or postgraduate students in the field of computer science. Seven of the ten participants rated their computer literacy as advanced, while the rest rated the computer literacy as intermediate. Therefore, all of the participants had some sort of knowledge and experience of human-computer interaction.

As mentioned in chapter 6 “Test design”, all of the participants had to perform all three of the scenarios, with both absolute semantic and context. Afterwards, the interview was conducted. Each session was lasting for 45-60 minutes, where the actual interaction with the spoken dialogue system took around 15 minutes. The briefing of the experiment procedure and the voice training normally took around 15 minutes, and the rest of the time was spent on the interview, 15 - 30 minutes.

7.2.2 Scenario 1: Time context

In the first scenario, time context, the participants experience how natural or unnatural it was to turn on the television with absolute semantic commands and context-dependent commands. The participants were also to rate the usability from very unnatural to very natural. The scenario was designed in such a manner that the absolute semantic command would not differ much from the context-dependent command. Even though, as
stated in figure 7.1, there was not that much of difference between the
inputs. The majority thought it was both more useful and natural to
use context-dependent commands, rather than using absolute semantic
commands. When participants were to rate between one to six how natural
the interaction was they rated 3.7 on absolute semantic and 5.2 on context
(Differ 1.5). When asked to do the same on usability, they rated 4.0 in
absolute semantic and 5.4 on context (Differ 1.4).

One of the participant, P2, thought that the scenario was somewhat
misleading because it seemed like the system took the participant’s feeling in
regard. Due to this, the participant felt uneasy with the scenario which made
the scenario seem less natural to the participant. P2 stated the following:
“I think that absolute semantic was the most natural in the first scenario.
I don’t think I could have an electronic device tell me when my favorite
television show is starting and that it told me what I felt. It took my
feelings in regard.”.

7.2.3 Scenario 2: Location context
The goal of the second scenario for a participant was to start a lamp in
the environment with the voice. The participants were informed that it was
getting dark in the office. The participant could then command the system
to turn on a light. When using absolute semantic, the participants had to
state their location as well. In this case, their location was the office. When
the participants were using context they did not have to state their own
location, because the system already knew their location.

The second scenario is regarded as more complex scenario in compared to the first scenario (see figure 7.2). The scenario does not only take the context into consideration, but also the location of the participant. Participants rated absolute semantic 3.1 in how natural the interaction was. The participants thought that context was more natural and rated it as 5.4 (Differ 2.3). The usability was rated 4.2 for absolute semantic and 5.7 for context (Differ 1.5).

7.2.4 Scenario 3: Event notification context

The third and last scenario placed the participants in an email scenario. The participants were informed that an email had been received. The participants could then command the computer to read it. Afterwards, the participants were given a choice to reply to the email and speak their own message. This scenario was regard as the most complex scenario, mainly because the participant had to give the system more information with absolute semantic in comparison with the other scenarios when using absolute semantic. The participant rated the scenario (see figure 7.3) to be 3.0 natural with absolute semantic and 5.3 with context (Differ 2.3). The usability was rated as 3.8 with absolute semantic and 5.5 with context (Differ 1.7).

One of the reason stated for why the third scenario was regard as less natural with absolute semantic commands, in contrast to the other scenario,
was because it was more complex. The participant had to state more information, in order to execute a command, in this scenario than in any other scenario. Context had a greater effect on the dialogue because of this. With context, the participant did not have state much previous information, as they would with absolute semantic commands.

7.2.5 Natural

![Figure 7.3: Results for scenario 3](image)

![Figure 7.4: Natural](image)
In the interview part of the test study, the participants were asked when the dialogue became more natural (see figure 7.4). Eight out of ten participants stated that it became more natural with context. The two remaining participants said that they did not experience any difference between absolute semantic commands and context-dependent commands. No participant stated that absolute semantic commands were more natural than context-dependent commands.

P6, one of the participants who stated that the dialogue became more natural with context, said the following: “It felt more as if it was a real person who understood you” and also said that the response became shorter because the participant did not have to repeat information given from the computer. According to the same participant that is how the interaction should be: “That is how it should be: short and natural”. P3 agrees with P6 and said the following about the subject: “When you’re not using context, it feels more like you repeat the question from the computer, which is quite unnatural because it’s not how you usually talk to a human”. P10 also thinks that it became more natural with context. Especially on the last scenario. When using context in the last scenario, the participant stated that it was like talking to another human: “That is how I would talk to a human”.

Two participants, P4 and P5, stated that the dialogue did not become more natural with context. P5 said that it did not make any real difference, because the user still had to say specific keywords. Even though P5 did not think it became more natural, the participant thought it became easier with context and stated the following: “... it was easier when it was a shorter sentence. In this way, the context was easier because there were usually fewer words that would be included.”.

P5 explains that the participant did not expect the dialogue to become natural. The participant was aware that the interaction was still with a computer and because of this it was not meant to be natural. P5 explains it as followed: “I’m not expecting it to be natural. Natural conversation for me is when you are talking longer and not just a really short or using keywords or something. That thing with explaining. I will never expect to explain to a computer, for now at least. I still think that they are just passive objects. They just do something when you tell them. Even if they tell you an event occur, you did tell them in advance in case of this event occurs, inform me of this. You always have to tell them. That is why I do not even think it should feel natural.”

The other participant, P2, who thought that absolute semantic commands were more natural over context-dependent commands had a hard
time to accept that it was a computer the participant was talking to. P2 was reacting negatively to the first scenario, time context, where the computer tells the user that the participant’s favorite show is starting. The participant regarded this as unreasonable and therefore unnatural. The participant was mainly reacting to the first scenario and thought the rest of the scenarios were more natural with context. The participant stated the following about the subject: “It was probably because it did not take my consider my feelings.” P2 was not the only participant who reacted to the fact that the participant was interacting with a computer. P2 felt that it was more natural to speak with a computer in the third scenario, event notification context scenario, because of the terminology. When there exist “computer terms”, then it becomes more natural with absolute semantic commands, rather than context-dependent commands. Even though it was not more natural to use context in the third scenario, it was more convenient. P2 stated the following when describing absolute semantic commands for the third scenario: “But it’s not as convenient. It feels natural to a computer, in some sort of way.”. Even though P6 thought that it became more natural with context, the participant said that sometimes it felt unnatural to interact with a computer in that manner. The participant explains that it becomes more logical to talk to a computer when using computer terminologically. It was more logically to use absolute semantic in the last scenario because of the terminologically but not more convenient accordingly to P6. Accordingly to P7, even if it is rather unnatural to interact with a computer in such a manner, the user will eventually get used to it. P7 states the following about the matter: “Humans are adaptable. Pure, deep down, we have not adapted with the phones because we still go around and check out the window to try to locate the other. But that definitely shows that we can adapt, even if it is highly unnatural. An unnatural phenomenon simply.”

7.2.6 Scenarios

One thing that became clear when interviewing the participants was that context-dependent commands are very well suited in some scenarios and not as well in others. The participants P5, P7 and P8 agreed that context-dependent commands are most suited in complex scenarios where they improve the interaction between the user and the system. The most complex scenario in the test study were the third and last scenario, the email scenario. Compared to that scenario to the first scenario, the time context scenario, which is the simplest one, the difference between the context and absolute semantic is higher. In the natural measurement, the difference between
context-dependent commands and absolute semantic commands in the email scenario is 2.3, while in the time context scenario the difference is only 1.5. The same also applies for the usability. In the usability measurement, the difference between context and absolute semantic in the email scenario were 1.7, while in the first scenario it was 1.2. P4 stated the following regarding the subject: “The most complex is the email scenario and I would assume that the context is more relevant there. But not to detract the television scenario either. It is a difficult question. I would say the email scenario”. The same participant answered the following when asked if it was because the scenario got more complex: “Yes, exactly. Then it will be more relevant with context”.

P3 agrees that there are scenarios that are more suited than other. P3 explains that it depends on what the user is doing at the same time as using the system. The participant goes on and gives a concrete example: “It is perhaps a classic example, but to sit with a phone and you have it in hands-free and driving and things like that, it’s very good so you do not hold the phone and look at and fiddle with fiddle with the screen.”. The participant continued describing why a car would be a suited environment for context-dependent commands: “That’s because it’s a direct threat to sit and hold on to a computer or a phone in a car, rather than to sit at home, because then you can sit and keep on fiddling with it without things might happen. You have to have a focus on something. It is perhaps more generally, in a position where you have to have full focus on your computer or maybe the thing you use. You would like to use it just as if talking to a friend beside you.”

All of the participants also agreed that there exist scenarios where context are not suited. One of these scenarios were in situations when there is no room for misunderstanding. In the following, P3 describes when to not use context: “...when you want to know that something will be done exactly as you want it to be done. So you ensure that this word must lead to exactly the event.”. P4 agrees with P3 and mention a concrete scenario: “Where you need to be very clear. The first thing I think of is a doctor.”. P4 continues and explains that it is easy that there could be some misunderstanding between the user and the system: “It can even happen when you and I are talking to each other. Could it happen when you and I are talking, the it’s even more likely that it could happen when I’m talking to a computer.”

P4 means that absolute semantic are necessarily. In some cases, the user is starting the conversation and therefore there exist no context to go on. P4 stated that context appropriate when there exist a dialogue and the user
wants to follow-up on an event: "If follow-up is necessary on something. If it is I who initiates it, for example 'Turn the lights on', there is no context to initiate from and it feels not as valid.". P5 also agrees that absolute semantic are needed, but said that you only should use them when it is necessarily. Otherwise the participant thinks that context should be used.

7.2.7 Usability

When did you experience that usability was at its highest?

![Chart showing usability preferences]

Figure 7.5: Usability

The usability was at the highest when using context-dependent commands in all scenarios according to survey conducted during the test study (see figure 7.5). When asked when the participant experienced the highest usability, seven out of ten participants answered with context-dependent commands. P4, P6 and P9 answered that they did not experience anything difference between absolute semantic commands and context-dependent commands. P6 explained why there was no difference: "It was high on both actually. With absolute semantic it is also very logical. The logical answer, but it is not very user friendly. I think they had the same (usability), because both were equally logical, but not as natural."

The lowest difference between the usability when using context-dependent commands and absolute semantic commands in all the scenarios were in the first scenario, TV scenario, with only 1.4. The highest difference occurred in the last scenario, event notification context scenario, with 1.7. One of the reason for why the last scenario had a higher usability with context-dependent commands according to P2 was that it went smoother with
context, in contrast to the absolute semantic were the participants had to state detailed commands to achieve their goal. The participant said when asked the second question: “With the context, because it was much smoother. With absolute semantic it became little contrived to repeat everything it told me. It wasn’t natural to repeat everything it told earlier. It is still very usable to be able to say 'Turn on the lights in the office' as 'Turn the lights', but not as useful as when the computer already knows what I’m already refers to”.

According to P1, another reason is that it became easier to learn how to interact with the system by using context: “In the context scenario its more natural way to interact with the computer, I think. I think that in the context it is easier to learn, because it’s the more natural way to do it”. P3 agrees with P1 and says this when asked if it became easier to learn with context: “Yes, because you did not have to remember the exact syntax how to speak”.

P5 also agrees that context increased the usability and stated the following: “I want to give the minimum words considering that I’m, afterall, interacting with a computer. I don’t really want to make chit-chat with it.”. P5 explains that you should not have to interact with the system more than necessarily. The answer from the user when the system triggers an event should be short accordingly to P5. This is also consistent with what P8 said: “I want to get what I want, quick and easy”.

### 7.2.8 Privacy

Before the participants started with the test study they were briefed about privacy and thereby informed that privacy was one of the subjects regarded in the study. Even though they were briefed about privacy, none of the participant felt that they were more observed or monitored when using context-dependent commands.

Six participant, P1, P2, P3, P6, P7, P10, all agree that there could exist a possible threat, but it all depends on how the data were treated. P1 raises the awareness that the context is saved by the system and could therefore cause a security risk. This risk would not be existing with absolute semantic commands, because the system itself does not need to store information regarding the dialogue with the user. P1 also states that the solution to this problem could be to erase the data after the session or after a period of time. P1 said that a period of 10 minutes would be acceptable.

P7 discusses the privacy risks of global information access. Due to the fact that the system is controlling private information regarding the user
and the environment, it is important to control the information flow. P7 state that the participant would be more secure if the data was only stored locally and transmitted in the local network: “I would like to know where all the data were stored. Would it somehow be collected on a device that has a connection, then I known that it is not optimal from a privacy aspect. But it would be stored locally, without connection, so I’d be more comfortable with it.”

P8 agrees to some extent with the previous participants regarding the security risk, but emphasises on which context the user is in. When asked if context could introduce a security risk, the participant said the following: “It depends on the current context. If I sit at home in my home and have this type of technology, absolutely not. If I stand at an ATM and it allows me to act in this way (using context)”. This statement is both closely connected to privacy question, but also the scenario question and raises the question on which scenarios are context suited.

One participant, P4, did even suggest the opposite regarding the privacy threat. The participant explained that there could be even less of a privacy threat with context-dependent commands, in contrast to absolute semantic commands. The reason for this is because the user has to send all information at once when interacting with the system. P4 stated the following: “I think that because of the absolute semantic commands will send more information. Send 'I want to watch TV on channel five,' while on the other it was just 'show it'”. By sending all information at once, there is a risk that there could exist eavesdropping when transmitting the information. But, if the eavesdroppers would listen to the whole dialogue between the user and the system, then the eavesdroppers would be able to place all the transmitted information into context. Perhaps context-dependent commands could provide another obstacle for eavesdroppers.

### 7.2.9 Participant choice

At the end of the interview the participant could choose which kind of command they would prefer. Only three participant, P6, P9, P10, choose context-dependent commands as their response to that question. The three participants all agreed that it became more natural and more human-like with context-dependent commands. Those were the reasons for choosing to use context-dependent commands, rather than absolute semantic commands.

The other remaining participants stated that they would prefer a combination of the two command types. P2 said the following regarding
the question: “... if you would have this kind of system I guess it would be a combination of context and absolute semantic.”. The same participant explains that the reason for choosing both is because it does not seem possible to interact with a system with only context-dependent commands: “I think that if it really was a great context, I think that it would always be better than the absolute semantic. But I do not know if that is realistic.”

P3 agrees with P2, and explains that it was more natural and easier to interact with a system using context-dependent commands. But the participant also agrees that absolute semantic commands are essential, because otherwise there is a risk of losing control. P5 says that there exists a risk of being misinterpreted when only using context-dependent commands. To eliminate the risk of being misinterpreted the user need to be able to use absolute semantic commands. P5 stated the following: “...you probably need absolute semantic more often than I want because you have too many devices”. Another reason for using both types of commands according to P8, is because they are suited for specific scenarios. The participant state that it all depends on the context.

One participant, P7, also said that it became more natural and easier with context-dependent commands. But because the participant had a problem with the voice recognition software when using context-dependent commands, the participant felt that absolute semantic would be better. The ASR interpret the word “it” sometimes incorrectly when the participant said the word. Even though the participant was having some problem with context-dependent commands, the participant admits that it would be better with context-dependent commands if they were functioning flawlessly.
Chapter 8

Discussion

The following discussion tries to draw conclusions based on the results collected from the test study combined with the literature study. The discussion also reflects on the created technology and the utilized methodology.

8.1 The impact of context

The conducted research had the underlying assumption that introducing context could affect the dialogue between the user and the spoken dialogue system and make it make natural and increase the usability. A spoken dialogue system is a specific type of an AmI system, and according to Michael McTear[8] it has two essential requirements: context awareness and natural interaction. The assumption of the research was that context-dependent commands could decrease Donald A. Norman’s[10] Gulf of Execution and make the interaction more natural and comprehensible. Based on the test study, context has the potential to increase the usability and make the dialogue more natural. According to the conducted survey, context-dependent commands are more natural and usable than absolute semantic commands. The survey also indicates that context has a varying effect based on the scenario. This is also supported by the results from the conducted interviews.

The first scenario was regarded as the least complex scenario and there were not many differences between absolute semantic commands over context-dependent commands. The difference between the two commands when measuring how natural the interaction was only showed 1.5 difference. The usability was only 1.4. When measuring the same on the last scenario the results were 2.3 and 1.7 respectively. One of the reason for these results
could be that the last scenario was considered the most complex scenario and therefore context had a greater impact on the interaction. It became rather hard for users to interact with the system when the user had to be very specific and use specific terminology. Using a specific terminology could be unnatural for a user because the user perhaps do not use it everyday. It also became unnatural for the user to constantly repeat what the computer already informed the user about.

Humans tends to use context in the dialogue when interacting with each other and therefore it would be more natural if the user could interact with the computer in a similar manner. Mark Weiser envision a future where computers were forced to fit into peoples' environment, rather than forcing people to enter the computer's environment. By introducing context into the dialogue we perhaps could be one step closer to achieve Weiser's vision. It could also make the “digital divide”\cite{6} smaller and allowing people who were restricted to interact with the computer because of the complex interface.

According to the conducted test study it became more natural with context for the majority, eight out of ten, of the participants. Context also increased the usability for the majority, seven out of ten, of the participants. The interaction became more natural mainly because the participants did not have to state as many keywords as they did with absolute semantic commands. This also increased the usability because it became easier and faster to give a command. Some participants felt that they were repeating the question when they did not use context. With context, the participants did not have to repeat as much information when using context in contrast to absolute semantic. The result of this made the interaction more human-like, and therefore more natural.

Even though the results are promising, there was still some participants who thought that context did not have any impact on how natural the interaction was. Accordingly to some participants, there were two reason for this. The first reason was that the participants felt that they still had to say specific keywords in order to execute a command and because of this there was no difference between context-dependent commands and absolute semantic commands. This of course is rather hard to avoid, because we still need some sort of keywords to specify want we to accomplish. Otherwise, we would have nothing else to go on, except context. Furthermore, one of the participants who thought that context did not make the dialogue more natural agrees that it still had some impact. According to the participant, it became easier to interact with the system when using context-dependent commands. The reason for this was that the participant did not have to state as many keywords as with absolute semantic commands.
8.1. THE IMPACT OF CONTEXT

Surprisingly, the second reason for why some participants felt that context did not make the interaction more natural, was because they were interacting with a computer in a natural manner. They could not accept that they were talking to the computer as they would with a human.

One participant stated an interesting point regarding the matter. The participant was not expecting the dialogue to be natural even to begin with. According to the participant, the participant knew from the beginning that the interaction was going to be with a computer. Because of this the participant could not accept the interaction to be natural. The participant stated that there has to be a longer dialogue between the user and the computer for it to be perceived as natural and similar to human-human interaction. Even though this could be a barrier for some users, one participant said that this is just another step of accepting new technology. Once the technology becomes common, the users would probably have an easier time to accept the technology.

All participants agreed that context-dependent commands are not suited in all scenarios and that there exist scenarios where absolute semantic commands have a greater effect. Context-dependent commands have a great effect in scenarios where there is a clear context and there is no risk of being misunderstood. If the context becomes rather unclear, there is a risk that the computer will misinterpret the user and execute the wrong command. There are also some scenarios where the participant might not want to risk being misunderstood and therefore choose to use absolute semantic commands, rather than context-dependent commands. Therefore, it is also necessary to have absolute semantic when the context is none existent, unclear or lacking. There was mixed opinions regarding when to exactly use context-dependent commands and absolute semantic commands. Some participants thought that complex scenarios were more suited to use context-dependent commands, while other thought the opposite. The results from the survey agrees with the participants that thought that context-dependent commands were more suited with complex scenarios. There were also some participants who provided concrete scenario for when to use context. One of these scenarios was in a car when driving. Because of the dialogue became more natural and easy to use, it would be suited in scenario where the user have to be concentrated and keep focus. One participant even stated that it as talking to your friend sitting beside you, rather than a computer.

As mentioned before, in the background chapter (see page 7), there are five different spoken dialogue system characterized by Michael McTear[8]. In the following list it is stated which types could be suited to use context-dependent commands based on the results from the test study:
Voice control: This type is suited for using context-dependent commands, because it allow the user to interact with the system for a longer period of type to control the environment.

Call routing: This type is not suited for using context-dependent commands because of the lack of context. When using call routing the user has to be precise, and therefore, it is need to use absolute semantic commands.

Voice search: This type also requires the user to be precise to accomplish correct search. But if there exist a clear context, than it should be possible to use context-dependent commands for voice search.

Question answering: This type also requires the user to be precise, otherwise there is a risk of the question being misinterpreted. But like the previous type, this type could also possibly function if there is a clear context.

Spoken dialogue: This type could be suited to use both context-dependent commands and absolute semantic commands. The systems functionality could both executed with and without context. When the user wants to be specific or there is no context, the user could use absolute semantic commands. The user could use context-dependent commands if there is a clear context and there is no risk of being misunderstood.

8.2 Privacy threat

Accordingly to Philip Brey[17], privacy is one of the biggest challenges for AmI. Because of this, the test study was focusing partly on the privacy issues that context potentially could have on the privacy. One of Marc Langheinrich’s[24] properties of an AmI system is memory amplification, which is described as the functionality of an AmI system to record users’ actions. The IUI technology also introduce another property, which is profiling. These two properties were implemented by introducing context into a dialogue between the user and the system. The system has to record previous information about the user and keep a profile in order to keep context. Otherwise the system would not be able to comprehend what a specific user means when using context references.
8.3. REVIEW OF THE DESIGN AND IMPLEMENTATION

With this in mind, the participants were asked in the test study if they felt more monitored. Even though the participants were briefed about privacy, none of them said that they felt more monitored. One reason for this could be that the participants were not used to the thought that the system was memorizing the dialogue. Perhaps the user would consider this thought if the system was implemented outside the test environment and into a real-world environment.

The participants may not have felt more monitored in the test environment, but six of the ten participants agreed that there could exist a potential privacy threat. Data management and storage was one of the most common privacy threats mentioned by the participant. With context-dependent commands the system has to store information in order to keep a context, otherwise the user would not be able to refer to the context. In order to handle this privacy threat, the data must be locally stored and preferably encrypted. According to one of the participant, the data should not be stored for a longer period of time. Because of system is handling vital information about the user, the data should be cleared after a period of time or after an interaction has ended between the user and the system. This result seems to agree with Aarts and Ruyter’s[5] three basic question regarding collecting data: Data Collection, Data Types and Data Access (see page 18).

One rather interesting point mentioned by one of the participants was that context could make the interaction more secure. The reason for this is because the user do not have to state all information at once. Therefore, context could prevent eavesdropping or make it harder. Without the context, eavesdropping the user’s commands could make it harder for the eavesdropper to make sense of the command. If the eavesdropper wants to contain all the information, including the context, the eavesdropper has to eavesdrop on both the input and output from the user.

According to Kaufman et al[11, p. 48], it is almost impossible to prevent eavesdropping. One method to solve this issue is to introduce cryptography and encrypt all the transmitted data. This will insure, to some extend, that no data will be interpreted by unwanted eavesdroppers. Introducing context could perhaps furthermore increase the security in that regard.

8.3 Review of the design and implementation

Both the design and the implementation of the spoken dialogue system worked as intended. The choice of using Dragon Dictate as the ASR proved
to be good. At the beginning of the study there was a fear that the voice recognition software would act incorrectly because of the different accent of every participant. Surprisingly the software did very well to understand certain dialects, such as chinese or swedish. Only at one point it failed to recognize a participant, but the participant was mumbling and speaking inconsistently which made it hard for the ASR to adapt and learn.

The choice of implementing the system in Java also proved to be valid. By implementing the system in Java, the author could easily use and integrate predeveloped code libraries to control the various devices used in the test study.

One critique that was brought up in the interview section, regarding the system, was the feedback for when the user spoke. The ASR sometimes took a while to analyse the input from the user, and therefore the user could not immediately get a status feedback on the recognition process. In some instances, participants started to talk before the ASR had a time to analyse the input, which were putting more effort on the ASR. Another critique regard the same subject was that the wait time before the ASR sent the result to the interactive system. Most of the time the ASR recognized the speech immediately and therefore the participant did not need the extra time to modify and correct the input. At the same time, the wait time was too low for a few participant. It is better to have the participants to wait some time, rather than having them experience stress.

8.4 Review of the Methodology

A qualitative method was regarded as an appropriate choice for this research. It was assumed that a conclusion could be drawn based on the results regarding the user’s experience and how context affects the interaction between the user and the system. The qualitative data allows me to develop a deeper understanding and support the hypothesis.

The initial brief instructions for the users were seen as suitable by the author. However, the briefing could be extended to further provide greater background knowledge for the participants. The briefing could be also extended with a test interaction with the system. The consequence of extending the initial brief could also confuse the participants. One participant was already confused with the terminology during the test study which affected the results. The participant mixed the meaning of absolute semantic and context-dependent.

If participants were provided with even more information, it could affect
the result both negatively and positively. The participant could also have been provided with more information how system interaction functioned, because some participants were surprised how the system was functioning at the first scenario. After the first scenario they had a better understanding on how it was functioning and could therefore use the system more effectively.

The entire test study was recorded and observed for further data. The observation did not provide any new data. One conclusion that could be drawn based on the observation was the problems some participants faced with the system. The ASR could not understand what one participant said and some participants felt that the system could have a better feedback. Both of these points were also brought up during the interview.

The conducted survey proved to be useful and help the participants to classify each scenario with absolute semantic commands and context-dependent commands. During the interview the participants could use the survey as a reminder to help them answer the interview questions. The survey also helped to draw a conclusion on how natural and useful each scenario was.

A semi-structured interview also proved to be suitable because it allowed the interviewer to ask the participants to extend their answer. This allowed the participants to evaluate their answer and provide a deeper understanding of the subject. The semi-structured mode also made the interview more natural and less linear. Because of this the participants could ask questions if there were any confusion regarding a question.

The data analysis method, content analysis, and the coding method both help to analyse the content and reach a conclusion. With these two methods, it was easier to get a deeper understanding of the results and their meaning.
Chapter 9

Conclusion

Introducing context into a dialogue, between a user and a spoken dialogue system, has proven to have the capability to make the dialogue more natural. The main reason for this is because the user did not have to state as much information in the context solution as the user would have to with absolute semantic commands. With context, the system has previous information to use when analysing the user’s commands. This has indicated to make the interaction seem more natural, mainly because it is similar to how humans interact with each other.

In some cases, context did not make the dialogue more natural. In some instances, the reason for this was because some users had a hard time to adapt to the fact that they were talking to a computer in such an anthropomorphic manner. The more anthropomorphic the dialogue became the less natural the interaction dialogue became. The ones who felt that this was an issue, also admit that this is a threshold to get passed and with some time it would become normal and natural.

It would also seem that context has the capability to increase the usability of a spoken dialogue system, but not in all possible scenarios. The usability was increased because the users did not have to state as much information, and therefore make it easier and faster to use. Because the interaction became more human-like with context, it was easier to understand and learn how to use.

One disadvantage with context-dependent commands is that they serve no function when the context is lacking or non-existing. Another problem that is introduced with context is the risk of misunderstanding. The risk of misunderstanding when one person is talking to another person, because of context, exist and therefore it could also exist when a user is interacting
with a computer. To compensate for lacking context or removing the risk of misunderstanding, the user should have the ability to use absolute semantic commands. With a valid absolute semantic command, there exist no risk of misunderstanding.

In this study, privacy did not seem to be an issue when using context-dependent commands. Even though context did not make users feel more monitored or aware of possible privacy issues, users agreed that there could be a theoretical risk of using context in a spoken dialogue system. The participants’ main concern was how the data was handled and accessed. According to the participants, users’ concern on the privacy would likely disappear if the user was aware of how the data was handled and that there was no risk of any vulnerable data being accessed globally.

Overall, introducing context to a spoken dialogue system has proven to have capability of making the interaction seem more natural and increase the usability in some scenarios. The results from this study indicates that context-dependent commands could have an impact on some of the usability attributes described by Jeffrey Rubin and Dana Chisnell[20, p. 4]:

**Efficiency:** Context-dependent commands could make the interaction between a user and a spoken dialogue system easier because of the usage of less keywords. This could increases the efficiency, which allows the users accomplished their goals quickly.

**Effectiveness:** Context-dependent command could have an impact on effectiveness because it enables users to use the system with ease and as expected.

**Learnability:** The learnability curve could become smaller with context-dependent commands because the users do not have to learn and memorize unnatural commands. This allows the users to interact with the system in a natural manner.

With these results, researchers could get a greater understanding on how and when to use context in spoken dialogue systems, and what sort of effect it has on the interaction between the user and the computer.
Chapter 10

Future Work

As a result of this study, future work was generated to further study the scientific area of spoken dialogue systems with context-dependent commands and absolute semantic commands.

- **Implementation of a spoken dialogue system in a real-world scenario**
  
  To emphasise the validity of the results from the test study, a spoken dialogue system should be embedded into a real-world scenario, such as the user’s home environment. However, the complexity of the implementation requires advanced preparations, such as implementing technology into the user’s environment and methods for the user to interact with them through the system. By doing this, one could validate the results on a larger scale and observe the full impact context has on the interaction.

- **Spoken dialogue system with context-dependent commands integrated into user environment:**
  
  To future study the impact context can have on the interaction between a user and a spoken dialogue system, the next step is to integrate the system into a user environment to test the long effect it has.

- **Integrated spoken dialogue system with context-dependent commands into cars:**
  
  One of the mentioned areas where context could improve the user’s usability is in a scenario where the user has to be concentrated and can not afford to lose the focus. The car scenario could be one of these
scenarios. In a car, a driver has to be constantly on alert and should not have to constantly think of commands to control the devices in the car. By introducing context-dependent commands into the interaction between the driver and the car could perhaps increase the safety and decrease accidents.

- **Could context-dependent commands introduce an obstacle for unwanted network eavesdroppers:**

  When investigating if context could introduce a privacy threat, one participant pointed out another interesting aspect of the issue. Transmitting information with absolute semantic commands contains often far more information rather than what context-dependent commands does. This leads to the question if context-dependent commands could create an obstacle for eavesdroppers, because they have to place the information into context in order to make any sense of it.
Bibliography


Questionnaire: Context-dependent voice commands in Spoken dialogue systems

The experiment that you are going to conduct today, is about introducing context to the dialogue between the user and the spoken dialogue system. A spoken dialogue system is an interactive system that has the capability to use speech as both an input and an output. These systems uses speech recognition to understand the user’s input and uses generated speech as an output. The study is focusing on whether or not context could make the dialogue between the user and system more or less natural. The study is also focusing on usability and privacy.

Please note that this study is not about speech recognition, but on context versus absolute-semantics in human-computer dialogue.

There are three different terms are importing to define before conducting this experiment:

- Natural: The term natural refers to when human-computer interaction resembles human-human interaction. The more human-like the interaction is between a system and a user, the more natural it is.
- Usability: The term Usability is the ease of use and learnability of a system. If a system is easy to use and easy to learn, then it has a high usability. If the system is hard to use and hard to learn, then it has a low usability.
- Privacy: Privacy can be defined as the right to be alone and the ability for people to seclude themselves or information about themselves. In computer science, privacy is often used when referring to preventing personal data from being collected and misused by others.

About you

1. Age: ______________

2. Gender
   - Male
   - Female

3. Computer literacy?
   - Basic computer skills.
   - Intermediate skills.
   - Advanced skills.

Scenario TV

Absolute-Semantic

Keywords: Start, Television, Channel, five
Example: Start the television on channel five

4a. Natural  Unnatural □□□□□□□ Very natural
4b. Usability  Unusable □□□□□□□ Very usable

Context

Keywords: Show, It
Example: Please, show it

5a. Natural  Unnatural □□□□□□□ Very natural
5b. Usability  Unusable □□□□□□□ Very usable
Scenario Lights

Absolute-Semantic
Keywords: Turn, On, Lights, Office
Example: Turn on the lights in the office
6a. Natural       Unnatural □□□□□□□□□□□□ Very natural
6b. Usability     Unusable □□□□□□□□□□□□ Very usable

Context
Keywords: Lights, on
Example: Turn the lights on.
7a. Natural       Unnatural □□□□□□□□□□□□ Very natural
7b. Usability     Unusable □□□□□□□□□□□□ Very usable

Scenario E-Mail

Absolute-Semantic
First interaction
Keywords: Read, Latest, Received, E-Mail
Example: Read the latest received e-mail
Second interaction
Keywords: Reply, Latest, Received, E-Mail
Example: Reply to the latest received e-mail
8a. Natural       Unnatural □□□□□□□□□□□□ Very natural
8b. Usability     Unusable □□□□□□□□□□□□ Very usable

Context
First interaction
Keywords: Read, It
Example: Please, read it
Second interaction
Keywords: Reply
Example: Reply to the message
9a. Natural       Unnatural □□□□□□□□□□□□ Very natural
9b. Usability     Unusable □□□□□□□□□□□□ Very usable

Notes
10. Participant’s notes:

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
Appendix B

Interview questions

• When did you experience that the dialogue became more natural: Using context-dependent commands or absolute semantic commands.

• When did you experience that the usability was at its highest?

• Did you feel more monitored when having context in the dialogue, in contrast to using absolute semantic commands?

• Do you feel that the context in the dialogue could cause a threat to your privacy?

• Which types of scenarios are suited for using context in dialogues and why?

• Which types of scenarios are not suited for using context in dialogues and why?

• Which types of commands did you prefer and why: Context-dependent commands or absolute semantic commands?

• Do you have any comments, remark on the experiment or the technology?
Appendix C

Data from survey

Description

The tables are structured as follows. The first column is the rate the participants could mark on the experiment i.e. 1 to 6. Rating 1 represents either unnatural or unusable and rating 6 represents either very natural or very usable. The second column displays the frequency distribution on how the participants rated on how natural the scenarios were. The third column displays the frequency distribution on how the participants rated on how useful the scenarios were. Example: Scenario TV - Absolute semantic, one participant rated 1 on natural, while four rated 5. One the same scenario, two participants rated 1 on usability and three rated 6.
Scenario 1: Time context

<table>
<thead>
<tr>
<th>Rate</th>
<th>Natural</th>
<th>Usability</th>
<th>Rate</th>
<th>Natural</th>
<th>Usability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>5</td>
</tr>
</tbody>
</table>

Table C.1: The tables presents the data from the tv scenario. The left table displays the data when using absolute semantic commands and the right displays the data when using context-dependent commands.

Scenario 2: Location

<table>
<thead>
<tr>
<th>Rate</th>
<th>Natural</th>
<th>Usability</th>
<th>Rate</th>
<th>Natural</th>
<th>Usability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>5</td>
<td>7</td>
</tr>
</tbody>
</table>

Table C.2: The tables presents the data from the lights scenario. The left table displays the data when using absolute semantic commands and the right displays the data when using context-dependent commands.
Scenario 3: Event notification context

<table>
<thead>
<tr>
<th>Rate</th>
<th>Natural</th>
<th>Usability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rate</th>
<th>Natural</th>
<th>Usability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

Table C.3: The tables present the data from the mail scenario. The left table displays the data when using absolute semantic commands and the right displays the data when using context-dependent commands.