Ubiquitous Computing in a Home Environment

Controlling Consumer Electronics

Tarik Hadzovic & Lars Thern

SONY

Master Thesis

Department of Design Sciences at the
Faculty of Engineering

LUND UNIVERSITY, SWEDEN

EAT 2014

Ubiquitous Computing in a Home Environment

Controlling Consumer Electronics

Tarik Hadzovic
hadzovic.tarik@gmail.com

Lars Thern
lars.thern@gmail.com

Department of Design Sciences at the Faculty of Engineering
Lund University

Advisors:
Günter Alce
Klas Hermodsson
Mattias Wallergård

June 5, 2014

Printed in Sweden
E-huset, Lund, 2014

Abstract

Building interaction prototypes for ubiquitous computing is inherently difficult, since it involves a number of different devices and systems. Prototyping is an important step in developing and evaluating interaction concepts. The ideal prototyping methodology should offer high fidelity at a relatively low cost.

This thesis describes the development of interaction concepts for controlling consumer electronics in a ubiquitous computing home environment, as well as the setup, based on immersive virtual reality, used to develop and evaluate the interaction concepts. Off-the-shelf input/output devices and a game engine are used for developing two concepts for device discovery and two concepts for device interaction. The interaction concepts are compared in a controlled experiment in order to evaluate the concepts as well as the virtual reality setup.

Statistically significant differences and subjective preferences could be observed in the quantitative and qualitative data respectively. Overall, the results suggest that the interaction concepts could be acceptable to some users for some use cases and that the virtual reality setup offers the possibility to quickly build interaction concepts which can be evaluated and compared in a controlled experiment.

Keywords: Mobile devices, Consumer electronics, Ubiquitous computing, Interaction, Service discovery, Prototyping, Context awareness, Virtual Reality, Augmented Reality, EASE Theme A

Sammanfattning

Att skapa prototyper för interaktion i en ubiquitous computing miljö är svårt, då denna miljö innefattar flera olika enheter och system. Då prototyping är ett viktigt steg vid skapandet och utvärderingen av interaktionskoncept, bör en bra prototyping metod bör erbjuda hög naturtrogenhet till en relativt låg kostnad.

Detta examensarbete beskriver utvecklandet av interaktionskoncept för användning i en ubiquitous computing hemmiljö. Dessutom beskrivs skapandet utav en utvecklingsmiljö, baserat på immersiv virtual reality, som möjliggör visualisering av nämnda interaktionskoncept. Konsumentelektronik samt en spelmotor användes för att implementera interaktionskoncepten samt för att bygga den virtuella miljön.

Statistiskt signifikanta skillnader och subjektiva preferenser kunde observeras i den kvantitativa respektive den kvalitativa datan. Sammantaget tyder resultaten på att inteaktionskoncepten kan vara användbara för vissa användare i vissa användningsområden. Den framtagna metoden möjliggör snabbt skapande utav interaktionskoncept samt utvärdering och jämförelse i kontrollerade experiment.

Acknowledgements

We would like to thank our advisors Günter Alce, Klas Hermodsson and Mattias Wallergård for making this thesis possible. Also, we would like to thank our colleagues at Sony and the Department of Design Sciences at the Faculty of Engineering at Lund University for their fantastic input. Last but not least we would like to thank Joakim Ericsson, Lund University, for his assistance as well as Ola Hedbäck, Lund University Libraries.

Tarik Hadzovic would like to thank family and friends for their support.

Lars Thern would like to thank his family and friends for putting up with him during the long winding road.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The aim of ubiquitous computing (UbiComp) is to bring computation into the physical world – seamlessly integrating people, devices and computation – using sensors, wireless networks and ubiquitous devices.

Imagine this:

> *While driving home after a long day at work, you are listening to your local news station. When you pull up to the garage door at your home, the doors automatically open for you. When you shut the car door the news broadcast is transferred to the nearest speaker. You walk inside your house which is unlocked and the alarm is disabled when you put your hand on the door. The news broadcast is again transferred to the nearest/ best speaker.*

The scenario above is an example of how people, devices and computation are seamlessly integrated in a smart ubiquitous computing environment and whilst it might seem futuristic, it is not that far away. The increasing number of smart mobile and wearable devices is set to change the way we perceive our surroundings and the way we interact with our consumer electronics. This chapter will give an introduction to the thesis work and present its purpose and goals. Also, the scope and limiting factors are defined.

1.1 Background

The modern day human owns and uses a lot of consumer electronics. In our home environment we surround ourselves with all sorts of devices: TVs, home entertainment systems, gaming consoles, digital photo frames etc. Each and every one of these devices uses its own set of interaction methods. They heavily rely on users' explicit actions: choosing what content to show, where to show it, how to show it and all the preferences adhering to the content.

To get into the right mindset, we need to consider what human - computer interaction (HCI) implies today. Also the latest development trend when trying to enhance the interaction between a user and his consumer electronics is discussed.

Explicit Human - Computer Interaction

The biggest reason why the above mentioned scenario could be viewed as futuristic is that it describes everyday actions in a way that is entirely different from how it is done nowadays. Today, when we wish to interact with our consumer electronics, we perform explicit actions: pushing buttons, manually connecting devices to each other and direct manipulation of graphical user interfaces.

If we were to describe the scenario from a today perspective, it would involve a set of explicit actions:

- Using a remote control to open the garage door or stepping out from the car and manually opening it.
- Unlocking the house door.
- Disabling the alarm.
- Turning on the nearest or best speaker to continue listing to the news broadcast.

As a consequence of the above described interactions you would miss a lot of the news broadcast. Also, you might have problems using your keys, remembering the alarm code and the right radio frequency.

The traditional way of interacting with consumer electronics is not very exciting and has been the same for a long period of time. In the current interaction paradigm, a high amount of explicit interactions is used, which leads to disruption, distraction and high workload for the users. Interactive systems need to be designed for greater degrees of implicit HCI. This would be in line with the ideas of UbiComp, discussed in Section 2.1.

There are too many apps for that

The latest trend in HCI is unfortunately in line with the current paradigm of explicit interactions - where each and every device has its own controller and specific interaction method. The interaction has become more virtual, in a sense that the explicit interactions have been moved into applications for smart devices. This does not account for a future where each and every device is seamlessly interconnected and connected to the Internet.

The increasing amount of devices humans use and environments they are used in leads to an ever increasing amount of applications, that need to be downloaded and managed in order to perform the most basic interactions. Clearly a paradigm shift is needed to be able to take charge of our interactions. To visualize how finding access to a service works in today's world a scheme was prepared, see Figure 1.1.

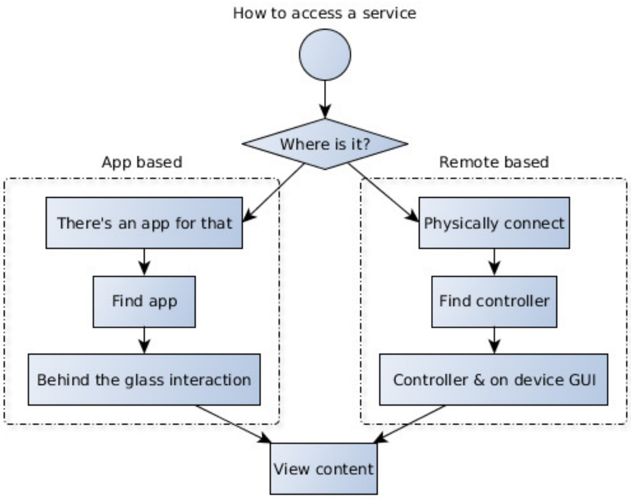

Figure 1.1: Decision making process to access a service

1.2 Purpose & Goal

The purpose of this master thesis is to:

- shine a light upon how realistic UbiComp scenarios can be prototyped
- show some use cases and benefits adhering to UbiComp.

New interaction concepts that support higher degrees of implicit HCI, aided by mobile and wearable devices, will be designed, prototyped and evaluated.

The interaction concepts aim to:

- simplify both discovery and interaction with consumer electronics
- yield a low workload for the users
- become more intuitive and natural to use.

Goals for this thesis is to challenge today's paradigm of explicit interactions with the new interaction concepts and to showcase them in interactive demos. The concepts will allow the users to interact with their consumer electronics in a manner that is not too novel - enabling users to easily start using the proposed interaction methods and prepare them for the coming paradigm shift.

1.3 Scope & Limiting Factors

The biggest limitation of this thesis is the restriction in man hours.

Another limitation of the scope is the fact that little thought is given to the technical feasibility of the concepts that were developed. As of today, there is no middleware available that can enable the interaction concepts that are proposed. However, nothing proposed is entirely ”science fiction”. None of the concepts, given the fast paced technical development, are impossible to recreate in real hardware and software solutions.

As a part of this master thesis the proposed interaction concepts were evaluated. A limiting factor in the evaluation was the somewhat small sample group possible to accommodate in such a short time.

1.4 Definitions, Acronyms & Abbreviations

In this section, expressions used throughout this report will be presented to give the reader a baseline.

Augmented Reality – AR

Augmented reality, is a technology that by Azumas definition [1]

1. Combines real and virtual
2. Interactive in real time
3. Registered in 3-D

The technology is used to superimpose virtual objects within a real world.

Consumer Electronics – CEs

Consumer electronics refers to any device that contains an electronic circuit board and is intended for everyday use by individuals.

Head-mounted display – HMD

A display device worn on the head is often refereed to as a HMD.

Human-computer interaction – HCI

Human Computer Interaction is the field that studies how to bridge the gap between computer systems and humans. The basis of the field is much dependent on computer science and behavioural studies. HCI considers:

- the consequences that arise from the interaction
- the consequences arising from failures
- how to make for attractive interaction methods.

Radio Frequency Identification – RFID

RFID is a technique to create identifiable tags, they can contain information from as little as a short identifier to a couple of kilobytes There are active and passive RFID tags, the active ones have their own power source and can send out their information over some distance. Passive RFID tags use the electromagnetic induction created when read to send out their information.

User Interface – UI

User interface is the part of a system that the user interacts with.

Wizard Of Oz – WOZ

Wizard of Oz is a methodology used mainly to prototype systems and perform usability testing. In WOZ non-existing parts of the system are simulated by a person.

Chapter 2

Theoretical Background

The purpose of this Chapter is to give the reader a better understanding of the fundamental ideas behind UbiComp. Related work is also presented.

2.1 Ubiquitous Computing

We live in a time of constant change. Technology, especially mobile technology, is developing at a raging pace. Our everyday mobile devices are getting smarter with every new generation – incorporating processing power with mobility. We surround ourselves with sensors and microprocessors and even wear devices with computational capabilities, such as visors[1], smart watches, smart wristbands etc. These devices could be crucial when creating ubiquitous home environments, that are context aware and support greater degrees of implicit HCI.

UbiComp is a multidisciplinary field of research, exploring computing technology and HCI beyond the desktop environment. It brings computation into the physical world, interweaving technology into our daily lives. UbiComp is believed to be the new era of human – computer interaction. This new era illustrates the *"one person, many computers"* - notion, as opposed to *"one person, one computer"* of the current PC era[2]. UbiComp relies on context-awareness and the implicit actions of users, rather than the context-free and explicit interactions of today.

Mark Weiser, the "father" of UbiComp, began his research at Xerox PARC in the late 1980's. He believed that the idea of a "personal computer" is misplaced and stated that *"Such machines cannot truly make computing an integral, invisible part of the way people live their lives."* and *"The most profound technologies are those that disappear"*[3].

Weiser's main thought is that computer systems should become invisible. Users should not be distracted from their tasks by needing to concentrate on a computer interface. Instead, they should be given the possibility to interact in a more intuitive and direct manner, in a context-aware environment, by talking, moving and gesturing.

[1] HMD, similar to a pair of glasses but with computational capabilities built in.

Poslad summarises Weiser's vision with the following three core requirements ([4] Chapt. 1.2.1):

1. Computers need to be networked, distributed and transparently accessible.
2. HCI needs to be more hidden.
3. Computers need to be context-aware in order to optimise their operation in their environment.

An additional two requirements are also proposed:

4. Computers can operate autonomously, without human intervention, be self-governed, in contrast to pure HCI.
5. Computers can handle a multiplicity of dynamic actions and interactions, governed by intelligent decision-making and intelligent organisational interaction. This may entail some form of artificial intelligence in order to handle:
   a) incomplete and non-deterministic interactions,
   b) cooperation and competition between members of organisations,
   c) richer interaction through sharing of context, semantics and goals.

Context-awareness and implicit HCI are two of the most important characteristics of a ubiquitous environment. These will be thoroughly discussed later.

Architectural Design

To structure the components of a UbiComp environment Poslad proposes the following model ([4] Section 1.4). In this model there are three basic parts:

- Smart environments – which follow the trend of embedding devices in the physical environment.
- Smart devices – which follow the trend of the ever increasing ability to manufacture low powered more complex devices with smaller footprints.
- Smart interactions – which follow the development of capabilities to create more interoperable and distributed devices.

Smart environments In smart environments, computations are made to augment ordinary activities of people situated in the environment. Smart environments consist of a multitude of connected devices, for instance WiFi access points, public screens and other consumer electronics. Coen defines smart environments [5] as:

> *"Intelligent environments require a highly embedded computational infrastructure; they need many connections with the real world in order to participate in it. However, this does not imply that computation need be everywhere in the environment nor that people must directly interact with any kind of computational device."*

Smart environments can enable implicit HCI without being obtrusive. If for instance the WiFi access point in a room makes a new connection to a novel personal mobile smart device, it could make itself prepared to share information about services in the environment to this device.

Since Moorse law, stating that processing power is increased twofold about every 18 months especially relative to cost and size, is applicable to sensors, embedded computers and active tags, the ability to cost efficiently keep track of the state in a smart environment will be possible.

In a smart environment one could see the benefit of monitoring states and information in the physical world. The windows of a room could sense if the light passing through them is strong. Should the light be strong and light sensors inside the room senses reflections in tables and computer screens the environment could act on its own and pull down the shutters.

Smart devices Smart devices can be either personal or shared. In the case of personal, this device can hold the users context and other relevant information about the user, such as login information for content providers and other services.

The personal device can be used to interact with and to show an arbitrary user interface. It could also be able to make for smart device/service discovery.

According to Poslad ([4] Section 3.4.1) the main characteristics of a smart device is:

- Mobility – users are able to move their devices into novel locations and connect to new services.
- Dynamic service discovery – users can find novel services and add their functionality arbitrary to the device.
- Intermittent resource access – they are upgradable and can be used in concurrency with other devices.

Smart devices are often of a multi-functional nature relaying a multitude of services to their users. A shared smart device, as proposed by Weiser [3], needs to be able to let an arbitrary user interact with it and to make use of its resources. A shared smart device could for instance be:

- A large screen used for collaborative work in an office environment.
- A public screen in the lobby of an organisation, letting users spontaneously and easily find locations/ book a meeting in the building.
- A tablet shared between family members, presenting different content/ services depending on who the current user is and letting them control other devices in the home.
- A ticket vending machine at a train station, simplifying purchase by trying to find information in the users context about where she is heading and proposing the correct ticket, eliminating the need to work through the UI.

Smart interactions Smart interactions extends the basic Synchronous/ Asynchronous interactions ([4] Section 1.4.3.2) with:

- Coordinated interactions, is when a multitude of devices/ services work towards a common goal, the coordination of the interaction could be centralised or distributed.

  If for instance using the example scenario mentioned in Section 1, the interaction would be centralised on a personal smart device carried by the the user, for instance a smart phone. This smart phone would know the users context.

  When the car comes into the driveway a sensor in the tarmac feels the car. Since the smart phone is situated in the car above the sensor, the smart phone coordinates a command to open the garage doors.

  Stepping out of the car and closing the door, the car tells the smart phone that the car is closed and locks itself. The smart phone now transfers the news broadcast to the closest/ most suitable audio source.

  When the user walks up to the door, the smart phone signals that a valid user is about to enter the house and gives the instructions to unlock the door and shut off the alarm.

  Walking inside, the smart phone knows that the user generally listens to auditory content on the main stereo in the living room and thus transfers the news broadcast there.

- Policy and convention based interaction extends the coordinated interactions but the interactions are based on previously agreed upon rules without the need of explicit interaction protocols. By setting up norms one can make the system act upon abnormal behaviour.

  An example could be if someone other than a valid user touched the doorknob in the previous example, an alarm could be sound.

2.1.1 Context Awareness

Humans are usually successful when conveying ideas between each other and making sense of the ideas shared. This is due to a large amount of shared social factors or shared context, like: *a*) shared language *b*) shared understanding about surroundings *c*) understanding everyday situations *d*) shared history. This is however not easily transferred into the realm of computers. To make for implicit interaction with computer systems, context needs to be shared with and understood by computers. This is done by context awareness and analysis.

Context in computers is built up upon large data sets and the analysis that can be made on the data. To explain contextual data Dey [6] says that: :

> *"Context is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."*

Systems that use context in order to make for implicit interactions are said to be context aware. Context awareness can according to Dey ([7] Section 3.1) be defined as:

> *"A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task."*

A scenario of how this could work is when a user enters a car that can connect with his smart phone. The *context aware* system would then aggregate the cars infotainment system with the users smart phone. The system might also degrade some functionality on the smart phone, such as playing games and other functions that might inappropriately distract the users. At the same time the system might augment the cars GPS system, by using the smart phones calendar to find where the user wants to travel and thus eliminating the need to enter information explicitly. The infotainment system would also have access to auditory content on the smart phone and calls would be forwarded via the infotainment system for hands free calling. For a context-aware system to be beneficial it should be able to [8]:

- *aggregate* and *disaggregate* third party systems seamlessly.
- *augment* and *degrade* functionality gracefully.

The above mentioned can be visualised in a transition scheme (Figure 2.1).

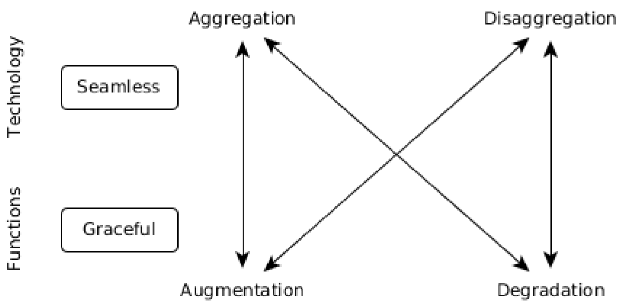

Figure 2.1: States of contextual aware systems[2]

[2] Inspired by [8,9] pg. 15

Benefits of context aware systems in the case of UbiComp systems are ([4] Section 1.2.4):

- Limits hardware and software resources to the ones that are relevant for the user, also cuts down on costs, since delivering all available services would be costly.
- Limiting access to unnecessary services, only those that are beneficial for the users shall be served.
- Minimizing the mental workload of the users to discover and select preferred services.
- Creating means for non intrusive decision making by the system on the users behalf.

Active or passive context aware systems

When discussing context aware systems there has to be made a difference between actively or passively context aware systems. If a system is fully active context aware, it will make the majority of the decisions on the user's behalf. This leads to loss of control. An example of this would be a location based service that always and automatically notifies all persons in a building when someone arrives.

There are some cases where a system should be fully actively context aware, for instance when security is critical or when there are tight time constraints. An example could be smart fire alarms which could forward notifications of fires to all nearby devices in the case of a fire.

Should a system be fully passive in its context awareness it would only report context to the user and make re distributions of the user interface. As an example, a navigation program could be said to be passively context aware, i.e. it knows when and where the user is and can show relevant information for the user based upon set preferences.

The goal would thus be to find a balanced point in between active and passive context awareness (Figure 2.2).

2.1.2 Implicit Human - Computer Interaction

Schmidt defines implicit HCI [10] as:

> *"... an action performed by the user that is not primarily aimed to interact with a computerised system but which such a system understands as input."*

The basic assumption here is that a computer has certain understanding of a user's behaviour in a given situation: it is aware of the user's context, *as discussed in Section 2.1.1.* This understanding is then considered as an additional input to the computer while doing a task. Implicit HCI is often regarded to be a complement to explicit HCI. Higher degrees of implicit interactions does however enable

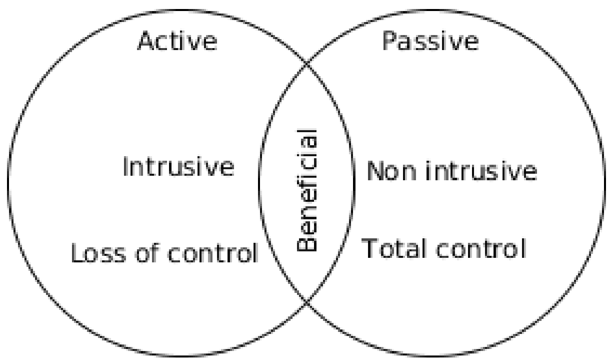

Figure 2.2: Finding the sweet spot

devices and technology to disappear, by hiding interactions from users. Implicit HCI is therefore a crucial part of Mark Weiser's vision for ubiquitous computing.

Poslad ([4] Section 5.1.4) mentions some challenges in supporting implicit interaction, namely the fact that it sometimes can be complex to accurately and reliably determine the user context because of:

- the non-determinism of the subject
- the user has not clearly decided what do to
- non-determinism in the subjects environment.

Furthermore, implicit interaction – in contrast to explicit interaction – reduces or removes explicit user control. Reducing the degree of explicit interactions in HCI requires a balance between several factors. First and foremost, users need to become more comfortable with giving up some or all control to automated systems, which could initially be perceived as obtrusive. Secondly, systems need to be designed so that they are able to both reliably and accurately detect the user and usage context while adapting their operation accordingly.

Implicit vs. Explicit Human - Computer Interaction

To additionally highlight the distinction between implicit and explicit HCI, Schmidt considers a smart garbage bin [10]. The garbage bin has the ability to scan the bar code of a product being thrown and composes the information for a suggested shopping list. A user performs the simplest of actions, throwing away an empty can into the bin, in the same manner as with any other garbage bin. The recognition abilities of the system (by scanning the bar code) as well as the built-in interpretation abilities (all items that go into the garbage bin may be on the next shopping list), make use of the action performed by the user instead of the user explicitly interacting with a computerised system.

This example emphasises that perception and interpretation are the key factors of implicit HCI. Figure 2.3 offers a schematic comparison between implicit- and explicit HCI.

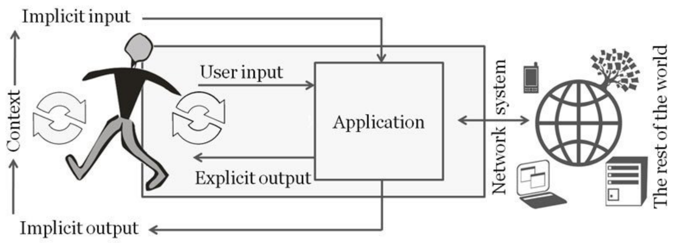

Figure 2.3: Implicit vs. Explicit human-computer Interaction[3]

2.2 Virtual Reality

Prototyping and developing UbiComp systems, in order to visualize complex, or somewhat complex interactions, is difficult. The methods and tools at hand today are inadequate. They are often expensive and do not offer the possibility to accept or reject proposed interactions at a conceptual level, early in a design process.

To be able to create interactive demos, showcasing the interaction concepts created during this master thesis, a virtual reality demoing environment was developed and used, as described in Section 4.2.

In this section different aspects of virtual reality (VR) are discussed.

According to Merriam Webster VR[4] is:

> *"...an artificial environment which is experienced through sensory stimuli (as sights and sounds) provided by a computer and in which one's actions partially determine what happens in the environment..."*

To better understand the realm of VR, some key concepts will be presented, namely presence and tracking methods.

Presence Lombard and Ditton [12] explain the concept of presence with six cornerstones:

- social richness - in what extent to which the medium is perceived as sociable, warm, sensitive, or personal when the virtual environment is used to interact with other people

[3]Picture taken from section 14.7 [11]

[4]Virtual Reality on Merriam Webster `http://www.merriam-webster.com/dictionary/virtual%20reality`

- realism - in what extent does the medium seem perceptual and socially realistic

- transportation - to what extent does the medium yield sensations such as if "you are there," "it is here," and "we are together"

- immersion - to what extent are the senses engaged by the mediated environment

- social actor within medium - to what extent do users respond to representations of other persons in the medium

- medium as social actor - to what extent is the medium itself perceived as a social actor, i.e. do users perceive the computer as a person

Schloerb adds two more definitions to the cornerstones [13].

- subjective presence - how great is the likelihood that the user believes himself to be physically present in the medium

- objective presence - how great is the likelihood of a user successfully completing a task in the medium

One can conclude the concept of presence as to how believable the system is. A system with extremely high presence will lead to the user in some cases not being sure of what is real and what is not.

Tracking In order to put a user inside a virtual environment, there is need for tracking input. With tracking, one would mean different input methods that take absolute and relative positions and rotations in the physical world and translates them to the virtual world. There are a lot of different tracking methods. The comparative Table 2.1 is built with data from [14] pg. 78.

Table 2.1: Comparison between tracking methods

Technology	Range(m)	Setup time(hr)	Precision(mm)	Hz
Optical: Marker based	10	0	10	∞
Optical: Marker less	50	0-1	10	∞
GPS	∞	0	5000	∞
WiFi	100	10	1000	∞
Accelerometer	10000	10	100	1000
Magnetic	1	1	1	∞
RFID:active	20 - 100	*when needed*	500	∞
RFID:passive	0.05 - 5	*when needed*	500	∞

Degrees of Freedom Different tracking systems have different amounts of degrees of freedom (DOF). The DOF can either be translational, rotational or both and it defines how many axes are tracked by the system. In Figure 2.4 the six possible axes are shown. Tracking systems referring to as having 6 DOF are tracking in all axes, whilst systems that only track in 3 DOF normally indicate if the tracking is translational or rotational. Some systems claim to be tracking in nine or more DOF. These systems normally consist of a multitude of tracking methods, each of them tracking in 3 DOF or 6 DOF, with the sensor data being fused for higher accuracy or speed.

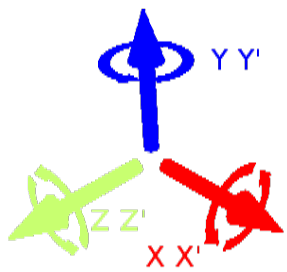

Figure 2.4: Cartesian axes

2.2.1 History of VR

VR is not something new. However, during the last couple of years a lot of technology enabling VR has become cheaper and easier to work with. One of the first HMD was created by Ivan Sutherland in 1968, and it was the next generation of "The Sword of Damocles", also presented by Sutherland in 1965. Whereas the first system was tethered to the ceiling, the second device could be worn.

During the 80's, NASA's Ames Research Centre created a interactive multimodal system [15]. The system became known as Virtual Interface Environment Workstation or VIEW. It consisted of a wide-angle stereoscopic display unit, a pair of gloves enabling input in multiple degrees of freedom (DOF), speech recognition technology, gesture tracking devices and speech-synthesis.

When Oculus VR started shipping their Oculus Rift Developer Edition (See Figure 2.6) in 2013, the subject of immersive VR exploded. Now developers could get a HMD with great tracking in 3DOF for 300$[5]. With the boom created by Oculus Rift, there were many other companies joining the development of technology for immersive 3D experiences. Existing technologies found new and impressive uses, with the input device Razer Hydra [17] released back in 2011[6] as one example.

[5] According to `https://www.oculusvr.com/order/`, feb 19 2014
[6] According to `http://en.wikipedia.org/wiki/Razer_Hydra`, 11 march 2014

Figure 2.5: NASA VIEW [16]

2.2.2 Pros & Cons

There are a lot of clear benefits when using VR as the means of testing novel interaction concepts on users:

- Reproducibility - When compared to other systems such as WOZ there is a high rate of reproducibility. When a certain test case has been developed, it can be re-tested as long as one has the necessary hardware available. External factors, when collecting data, are kept low since a test leader only has to focus on giving instructions.
- Cheap as in the monetary aspect - Since consumer electronics with tracking abilities becomes more and more common ware and mobile devices are getting more sensors and becoming yet more powerful by each iteration, there are soon no reason for proprietary systems. With software and hardware projects such as the OpenDive [18], a low end Android smart phone can be used as an see-through AR system or fully fledged VR system. They already have all the sensors needed for an immersive experience.
- Cheap as in time - With a Game Engine such as Unity [19], one can quickly develop an immersive experience. Many hardware manufactures for VR devices releases Software Development Kits (SDK's) for Unity and it is quick and easy to import 3D objects into the engine.

There are also some cons of using VR:

- Novelty - Since the idea of testing interaction concepts on users via VR is fairly new, there is not much research on how usability testing adopts to

Figure 2.6: Oculus Rift Head-mounted display

VR. There are no clear models for understanding what noise is introduced in the collected data from the system.

2.3 Related work

The subject of UbiComp all started with Mark Weisers paper The Computer for the 21st century[3]. The first words in the paper written in 1991 are:

> *"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it."*

These words were more or less a vision for a brave new world of possible future computer systems. Nowadays it has more or less become a necessity to explore the subject of UbiComp. We are standing at the gulf where IPv6 is taking over from IPv4. This will have the added benefit of higher and easier interconnectivity and scalability. There are more and more smart artefact's coming out each and every year and interconnecting them for human benefit must be a priority.

In this section a short look at some previous work done in the field of UbiComp is presented.

A Survey of Research on Context-Aware Homes This paper [20] is a survey of what context-aware homes could imply and what main points need to be further researched. It approaches the subject from the user's standpoint. They bring up some usage scenarios and put down a couple of requirements of context-aware systems in the home. In order to penetrate, the systems need to be:

a) safe *b*) supportive *c*) convenient *d*) pleasant *e*) enjoyable *f*) entertaining *g*) relaxing.

They also emphasize on these points being taken care of in order to penetrate: *a*) usability *b*) social acceptance *c*) privacy concerns *d*) low cost *e*) zero administration.

According to the paper the most important subjects for working towards context aware homes are:

- Instrumentation – All the building blocks of a Context-Aware system. Sensors, network infrastructure and new means of interacting (input devices) with devices.
- Middleware – Infrastructure on the whole, hardware abstraction from all possible sensors, context managers and new privacy managers.
- Applications – Software to act on the information gathered from sensors and input devices.
- User experience – Meeting users expectations and minimising frustration and at the same time make experience enjoyable.
- Privacy – Since a context aware system records a plethora of information about users, privacy needs to be taken into account every part of the way.

Evaluating user preferences for video transfer methods from a mobile device to a TV screen The article [21] evaluates four different methods of transferring video content from a mobile to a stationary TV screen. They found that the participants preferred interaction methods familiar to them. Strong factors in how methods are perceived are:

- familiarity – the more familiar the interaction method is, the more penetration it's likely to have
- convenience – the easier a interaction method is to perform the better
- annoyance – if a interaction method is error prone it's not perceived as a good method.

Outdoor augmented reality gaming on five dollars a day The paper [22] presents what they call an immersive HMD based, wearable 3D user interface that is kept inexpensive. They could achieve this since they saw the convergence between consumer electronics and virtual reality. They saw that there was not a need for proprietary systems any more.

Research on Presence in Virtual Reality: A Survey This paper [23] gives an overview in the field of presence, what concepts need to be accounted for in order to gain high presence in a VR environment. Most importantly they discuss the importance of more research into the field.

Formal Analysis of Ubiquitous Computing Environments through the APEX Framework APEX framework was used by Silva et al. [24] to evaluate ubicomp concepts. The framework offers the possibility to simulate complex scenarios and to work with models, which can be moved around in a virtual environment. APEX is based on OpenSimulator, which is mainly designed for online multi-user 3D worlds and interactions are performed in 3rd person perspective.

Chapter 3

Method: Human - Centred Design Process

The main goal of this master thesis was to develop a number of interactive demos, showcasing interaction with CEs in a UbiComp home environment. Before any implementation could take place, the interaction concepts needed to be designed. Also, to make for realistic visualisation of the interaction concepts with no regards to possible technical limitations, a suitable prototyping and demoing environment was necessary. To be able to achieve the goal, a human - centred design process was initiated. Figure 3.1 provides a schematic description of the four phases the process consisted of.

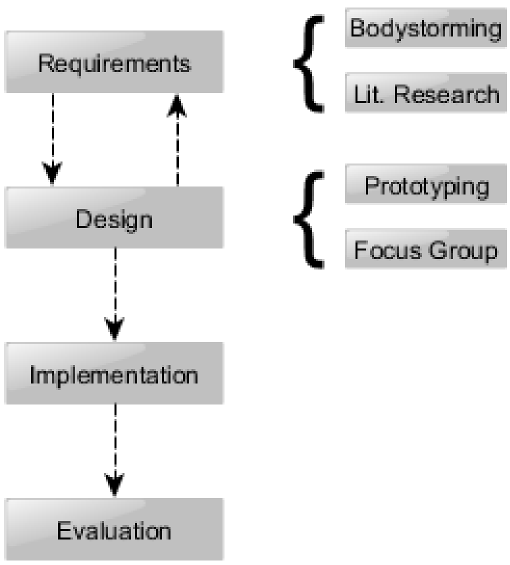

Figure 3.1: The overall process

3.1 Requirements phase

During the requirements phase, the objective was to define user and usability requirements. This ensured that all factors relating to the use of the proposed interaction concepts were identified before any design work would start. Also, it provided the basis for the usability testing, *described in Section 4.4*.

Through bodystorming and literature research, the initial context of use, some scenarios and constraints were defined.

3.1.1 Bodystorming

Bodystorming [25] sessions were conducted to broaden the approach to the subject and to be able to define the initial requirements for the interaction concepts. The idea behind bodystorming is to pretend a product, or in this case interaction, already exists – ideally in the place it would be used. Therefore, the bodystorming sessions were conducted in a home environment, with collection of CEs. Consequently, the first constraint was that the interaction concepts should be used within a domicile.

The thesis description stated that device discovery and device selection should be simplified, through the use of mobile devices such as a smart phone, a tablet, a smart watch and a pair of visors. This rendered in another constraint: the interaction concepts should simplify both device discovery and device selection, through the use of mobile devices.

Ahead of the bodystorming sessions three basic scenarios were defined:

- A user wants to start video playback on the TV set.
- A user wants to display photos on a digital photo frame and to print a document.
- A user has five minutes to spare and wants to spontaneously interact with one of the devices in the living room.

The starting point of the bodystorming sessions was an investigation on what interacting with CEs implies today. Performing bodystorming, the thinking was "outside the box" and creative. This trend was also noticeable in the second and third bodystorming sessions. The way interaction with CEs was performed was completely different from how it is done nowadays as movement, gestures and speech were the main interaction methods.

The bodystorming sessions were successful since they broadened the approach to the subject: thinking beyond the interaction paradigm of today and visualising how interaction could evolve in the future. At this point extensive literature research was needed.

3.1.2 Literature Research

Literature research was conducted to further investigate how interaction with CEs might evolve in the future. Wigdor's and Wixtor's book *Brave NUI World* [26] offered some early inspiration.

Throughout the literature research, well over a hundred books, publications, papers and articles, that relate to this master thesis, were skimmed through. All of these were saved and organised in a shared group library with the Zotero [27] reference manager, for later use.

When reading all of the material found, the term ubiquitous computing was stumbled upon. Ubiquitous computing seemed to be a perfect fit for the way interactions were performed during the bodystorming sessions. Poslad's book *Ubiquitous Computing – Smart Devices, Environments and Interactions* [4] was used as the basis for finding theories for the future interaction concepts to be developed. Also, it would be the basis of the theoretical background for the entire thesis work. To be able to define and motivate design choices made when designing the interaction concepts, Norman's *The Design of Everyday Things, Revised and Expanded (2013)* [28] was mainly used.

3.2 Design phase

The initial requirements for the interaction concepts were defined in the previous phase and the next step was to start designing and prototyping. Thoughts on future interaction that were accumulated during the bodystorming sessions and the literature research, were used as input to this phase. Between this phase and the previous one there were numerous iterations as new ideas for future interaction constantly emerged.

The main activities during this phase were *prototyping* and a *focus group session* and the aim was to finish the design of the interaction concepts and to create the prototyping methodology used to develop the interactive demos.

This section describes the method when designing interaction concepts and developing the prototyping methodology.

3.2.1 Prototyping

Since the goal of this thesis was to create interactive demos showcasing interaction concepts, a methodology for iterative prototyping of concepts was needed. The way interaction concepts were prototyped was to first create a Lo-Fi prototype in order to envision how the interaction could be performed. Based on the Lo-Fi prototype, a more interactive approach, based on immersive VR, was created.

Lo-Fi Prototyping

Previously conducted bodystorming sessions resulted in a number of requirements for the interaction concepts. A general idea of what the result should be was: *a context-aware UbiComp home environment that allows for high degrees of implicit HCI and which seamlessly connects users, devices and computation.* It was noticed during the bodystorming sessions that the key to creating such interaction would be allowing users to interact with their CEs in a more natural way. All of the above was used as input when producing the pen-and-paper Lo-Fi prototype, seen in Figure 3.2.

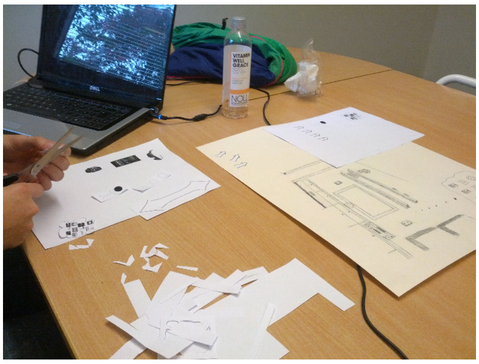

Figure 3.2: Creating the pen-and-paper prototype

The general feeling was that the Lo-Fi prototype was not enough to visualise the proposed interaction concepts. A more interactive approach would help to solve this issue.

Hi-Fi Prototyping

After the inital thoughts had been modelled in a Lo-Fi prototype a Hi-Fi prototype was needed. Different approaches were considered of how to create the interactive demos showcasing the interaction concepts: a WOZ setup, AR and VR.

In the beginning of this thesis work a number of devices that could help when developing the interactive demos were given: the Oculus Rift [29], Razer Hydra [17], LEAP Motion [30] and the Sony SmartWatch [31]. Going forward, focus was to be on creating a virtual demoing environment using a methodology based on immersive VR - where a character interacts with a collection of CEs. This seemed like the best approach considering the different demo tools at hand.

To be able to build a virtual demoing environment, the Unity Game Engine [19] was used. This could enable the visualisation of complex or somewhat complex interaction concepts, in a UbiComp home environment.

Multiple setups and technologies were tested out and the results of this can be read in Section 4.2.1 pg. 29. Some conclusions on the requirements for the demoing environment can be read in Section 4.2.1 pg. 31.

When the virtual living room had been created, the overall look and feel of the demoing environment, was set. In the next phase, the interaction concepts and the demoing environment would undergo final implementation.

3.2.2 Focus Group

Potential user comments and feedback is vital for the end result and therefore a focus group session was conducted. The focus group session consisted of two parts, one unstructured and one structured part. Previously, a Lo-Fi prototype was produced to visualise the early conceptual design. The prototype was used as discussion material in the structured part of the focus group.

Eight participants attended the focus group. Since the purpose of the focus group was to discuss interaction design and especially future interaction with CEs, the participants were carefully chosen. It was identified that the participants needed to have some prior knowledge of and interest in interaction design. All of the eight participants were first and second year students at Lund University, studying Information and communication technology, Computer science and Technical design.

The unstructured part of the focus group indicated that gestures and speech are the most popular methods of interacting with ones CEs in the future, thus validating the initial thoughts on the subject. Some very important points were made during the focus group, mainly regarding the possible issues with such interactions. The feedback given during the focus group, presented in Section 4.2.2 pg. 32, was analysed, noted and used to further improve the interaction concepts.

3.3 Implementation phase

This phase was initiated when the interaction concepts were fully designed. If implementations in the previous prototyping activity were done in a more of a exploratory manner, this phase strived towards making the demoing environment mature enough for the coming evaluation.

The final setup of the demoing environment is presented in Section 4.3.1 pg. 39. Also, the hardware used to create the demoing environment will be in this section. Last but not least all the software used when creating this demoing environment is listed in Section 4.3.2 pg. 41.

3.4 Evaluation phase

To evaluate the interaction concepts and gain information on usability issues, they were to undergo usability testing.

A test plan was created, and literature research pointed towards what measurements were interesting to collect. In order to do calculations on the collected data, it was necessary to read up on different analytical methods and what different program suites could be used to perform the calculations. The results and discussions adhering to the evaluation phase can be read in Section 4.4 and results and discussions adhering to the usability test is presented in Section 5 pg. 47.

Chapter 4

The design process: Results & Discussion

This chapter contains the results of the design process as well as discussions adhering to it. In Section 4.2.3 pg. 34 the final conceptual design is described and the design choices made are motivated. Also, in Section 4.4 pg. 42 the setup of the usability test is presented.

4.1 Requirements

When investigating the interaction paradigm of today the following was noticed:

- The disruptive and distracting use of controllers/ terminals as well as other interaction methods, when controlling one's CEs.
- The overwhelming amount of explicit actions needed to decide what content to display and where to display it, as well as when entering the users' preferences and manually connecting devices to each other.
- The lack of communication between devices but also between users and devices.

The way interaction with CEs was performed during the bodystorming sessions was quicker, in the sense that the steps from wanting to do something to actually doing it, were minimised. It felt much more natural and intuitive and controlling the CEs was done by moving, gesturing as well as talking and the CEs acted upon this. Consequently two requirements, for the interaction concepts were defined:

- Interaction should yield a low workload, by no or little use of controllers, apps and different UIs.
- Interaction should become more natural and intuitive, lowering the amount of explicit interactions needed when wanting to control ones CEs.

As mentioned in Section 3.1.1 pg. 22 two constraints were defined: *the interaction concepts should be used within a domicile* and they should *simplify both device discovery and device selection*.

The initial requirements set up during the bodystorming sessions were therefor set. The interaction concepts aim to:

- simplify both discovery and interaction with consumer electronics
- yield a low workload for the users
- become more intuitive and natural to use.

These requirements made up the basis for the usability testing, as presented and discussed in Chapter 5 pg. 47.

4.2 Design

This section describes the iterative conceptual design, development of the VR demoing environment, design choices made and the final conceptual design.

4.2.1 Prototyping

Pen-and-paper prototype

Two major interaction concepts, called *Grabbing to control* and *Intent by talking* were visualised in the pen-and-paper prototype. These concepts contained all the interaction methods thought of during the bodystorming sessions, described previously.

In **Grabbing to control** the main thought was to let the user control CEs by grabbing towards them. The CEs are all situated in a context-aware, smart living room: they are interconnected and can exchange information.

Device Discovery
To simplify device discovery this concept uses natural gestures and the benefit of having a computational device, the smart watch, attached to the user's arm. To discover devices situated in the smart living room, the user extends the arm with the smart watch attached, towards the CE that the user wants to interact with. The smart watch will vibrate to tell the user that the CE can be interacted with. Performing a grabbing gesture in the same direction will temporarily activate the CE and information about it are shown on the user's smart watch. The user can choose to either "throw away" the device or to control it.

Device Selection
Should the user be satisfied with the CE that was previously chosen, the user moves the hand over to a device (smart phone, tablet, smart table etc.) from which the user would like to interact with the CE. Performing zoom-like gesture on this device will expand a user interface corresponding to the CE previously selected.

In **Intent by talking** the main thought was to let the user control CEs by telling them what the intended action is. The CEs are all situated in a context-aware, smart living room: they are interconnected and able to exchange information.

Device Discovery
In *Intent by talking* there is no real need to discover what CEs are available. A user talks to the environment, the smart living room, through the smart watch and tells what is to be done. If it is possible to do it, the controlling capabilities will be made available. If not, the system will tell the user so. Also, a user who has no prior connection to a certain room/area containing a collection of CEs, can ask the systems what actions are available in this room/ area, through the smart watch.

Device Selection
When the user tells the smart living room what the intended action is, the intent will be matched to the actions that are located in the smart living room. Depending on who the user is, the user's intention and context, appropriate CEs will be made available and their information, UI and controllers will be available at the user's fingertips. Now the user can choose to act in some way. Since the user is wearing a pair of visors, an anti-pinch gesture can be performed on every surface. Now the chosen CEs controllers will be augmented and the user will be able to control the devices. If no appropriate device are available, the system will tell the user through the smart watch, enabling a dialogue-like interaction between the system and the user.

The two above mentioned concepts incorporate some strict design choices. These will not be discussed nor motivated, as the two concepts were later rejected. Some parts of the design were reused in the final concepts design and this will be discussed thoroughly in Section 4.2.3.

VR Prototyping

Multiple proposed setups The first thing tried out was to use the demoing devices together - meaning that every device would serve a unique purpose:

- The Oculus Rift was used to put users into the virtual environment, offering an immersive experience.
- The LEAP Motion (Figure 4.1) was used to track the user's finger movements and hand gestures.
- The Razer Hydra hand controller (Figure 4.2) was used to track the user's hand movements.
- A Sony SmartWatch was used for additional input, like swiping and tapping, as well as for feedback.

The list above above is the initial setup.

Some major limitations were noted, the biggest ones being: the inability to track the fingers when the hand goes outside the narrow field of view (FOV) of

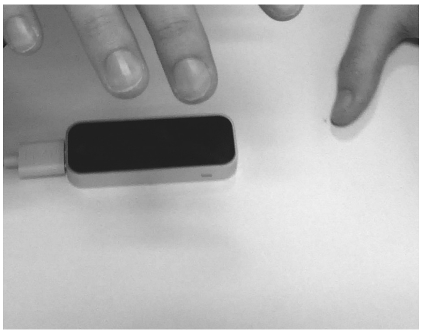

Figure 4.1: LEAP Motion

Figure 4.2: Razer Hydra

the LEAP Motion[1]. A better way of truly tracking the state of a user's fingers was needed. Also, the limited computational capabilities of the Sony SmartWatch - coupled with the difficulty to connect it to the computer running the virtual environment was a major limitation. Other limitations were, the short cables of the Oculus Rift and the Razer Hydra, the somewhat low resolution of the Oculus Rift and the inability to be hands free whilst tracking hand movements with the Razer Hydra.

Another setup was considered. The LEAP Motion as well as the Sony SmartWatch were ruled out. Instead, two new devices were introduced: the 5DT Data Gloves[32], for finger tracking and a smart phone to simulate a smart watch/ smart wristband. The main benefit of using the Data Gloves instead of the LEAP was that fingers always could be tracked. The new setup consisted of:

- The Oculus Rift - was used to put users into the virtual environment, offering an immersive experience.
- The Razer Hydra - was used to track the user's hand movements.
- A pair of 5DT Data Glove 5 Ultra (Figure 4.3) - were used to accurately track the user's fingers.
- A smart phone - was used for additional input, like swiping and tapping, as well as for feedback.

The new setup offered new possibilities and was a lot more robust. After the new setup had been decided upon, some focus was put towards tinkering with the virtual environment, to further improve the experience. A virtual character was created with Project Pinocchio (Section 4.3.2 pg. 42) and inputs from the demo devices were mapped to this virtual character.

Furthermore some thought was given on the possibilities to enable user speech input. An application for speech recognition, that could be run on an Android smart phone was developed. This solution was quickly tested and it was noted that this could be a possible way to implement speech input, given that the interaction concepts would require this. A major limitation of this solution was that

[1] The FOV of the LEAP Motion is 150 ° and the LEAP Motion worked best at the distance 3 cm - 20 cm of the object tracked

Figure 4.3: 5DT Data Gloves 5 Ultra

it was not sufficiently fast as both speech recognition and the communication between the smart phone and the computer running the virtual environment was rather slow.

There was only access to one Razer Hydra and since there was no access to other tracking devices for the user's elbows, shoulders etc. a decision was made to use inverse kinematics. Inverse kinematics would enable natural looking movement of the virtual characters arms, based on the input from the Razer Hydra. After experimentation and tests with different libraries for this purpose, one inverse kinematics package was chosen (Section 4.3.2 pg. 42).

Requirements for demoing environment All of the proposed concepts were to take place in a home environment. Thus the creation of a virtual living room was undertaken. There were some requirements for this living room, namely:

- The living room should look familiar.
- The living room should contain a number of CEs.
- The living room should be created so that a user can look around in all directions, as to learn how the Oculus Rift reacts to head movement.
- The living room should be created in a manner that enables the easy addition and removal of objects.

In order to encompass the requirements, the living room was created as a prefab. A prefab is a type of asset for Unity that enables re-usability. Should a prefab be changed in a scene[2], all other instances of the prefab will be updated accordingly.

As there were no plans on letting users walk around in the virtual living room, a restriction for the character to be seated in a couch was decided upon (as seen in Figure 4.4 pg. 32). The decision was also based upon that;

- movement inside the virtual living room was not important in order to gain high presence.

[2]A scene is what contains all objects to be included in a runnable file

- all the cables that were connecting the devices, restricted user movement to a circumference of about 0.3 meter.
- extensive movement inside the environment leads to cyber sickness, since the Oculus Rift has no position tracking.
- the arm tracking and the inverse kinematics used, was not robust when the test subject moved away from a sitting position.
- movement away from the sofa was not important to showcase the concepts.

Input from the conceptual design and the exploratory development of the prototyping environment decided what virtual devices should go into the demoing environment. Thus some different models of television sets, stereos and media devices were explored. After a couple of iterations with different devices, the final setup of the devices in the demoing environment was set:

- one large television set
- one game console
- one stereo
- one set of speakers
- one printer
- one tablet
- one smart watch/ smart wristband.

The result from VR prototyping can be seen in Figures 4.4 and also in Figure 4.5 and 4.6 pg. 35.

Figure 4.4: Character Charles in the virtual environment

4.2.2 Focus Group

As mentioned in Section 3.2.2 pg. 25, a focus group session was conducted to both validate the early conceptual design but also so gain additional input to the final conceptual design. The focus group session consisted of two parts, an unstructured part and a structured part.

Unstructured part

In the unstructured part of the focus group the participants were given the opportunity to freely discuss future interaction with CEs. To spark the discussion the participants were asked to discuss the interaction paradigm of today and to consider how it might evolve in the future. Quickly the discussion became mainly focused on two types of interactions, gestures and speech. The following points were made:

- Speech interaction is good if it is well implemented – it is beneficial to users since it is quick and precise.
- Today speech interaction is not a widespread interaction method, due to poor technical implementation and lack of social acceptance.
- Speech interaction is more natural to use than gestures, when a user has a specific intent. Using gestures, the user would first have to learn which gestures corresponds to the intent in mind.
- Controlling CEs with gestures can feel natural but in some scenarios it could also be perceived as ridiculous.
- A major issue with gesture interaction is the risk of false positives.
- Future interaction is probably a combination of gestures and speech as they complement each other well.

Structured part

In the structured part of the focus group the Lo-Fi prototype of the two interaction concepts, *Grabbing to control* and *Intent by talking* was presented to the participants. The concepts were explained and the participants were asked to give their feedback and to discuss possible issues. Many important points were made, mainly the following:

- A major issue with grabbing towards something you want to control, is the fact that you are grabbing towards something that is essentially virtual – this could be hard to understand for a novice user.
- Feedback is important, signifying the consumer electronics that can be controlled by grabbing towards could make sense.
- A major issue with augmenting virtual instances in the physical world is that the user could feel a bit disturbed if a lot is shown at the same time.
- The technical feasibility of the concepts is questionable. Creating apps for controlling the CEs would be much easier.
- False positives are a major issue. There is a strong need for some kind of start command for both concepts.
- Being forced to carry and wear a lot of different mobile and wearable devices in order to be able to interact with CEs is problematic, since it limits interaction.

- The smart watch will probably be the most successful wearable device.

The comments above were the basis for the final iteration of the conceptual design. They gave an insight into how acceptable the interaction concepts are to potential users. Possible issues were later addressed and design choices were carefully thought through in order to increase acceptance.

4.2.3 Concepts

At the end of the design phase, a plethora of interaction concepts had been developed. After a while the amount of interaction concepts was hard to manage. A table was created easier management. Many interaction concepts in this table were rejected. The table and the concepts are presented in Appendix A.1.

Creating the table of concepts lead to critical thinking. Eventually, four interaction concepts, two concepts for *Device Discovery* and two for *Device Selection* were designed. They are presented and the design choices adhering to them are discussed below.

In order to evolve the interaction between users and CEs, the interaction paradigm of today needs to be challenged. This could be done by creating ubiquitous home environments, that are context aware and thus allow for a higher degree of implicit HCI.

During prototyping (Section 3.2.1 pg. 23), it was identified that the end result should be: *a context aware UbiComp home environment that allows for higher degrees of implicit HCI and which seamlessly connects users, devices and computation.* The demoing environment developed throughout this master thesis enables the implementation of such environments, effectively enabling the visualisation of realistic future interaction scenarios. Here the different components of the UbiComp home environment are discussed.

Context Awareness and Implicit HCI Context awareness is one of most important components of a UbiComp environment. Making use of *user context* is a good way of enhancing a user's experience - making interaction more relevant, just-in-time and enabling implicit HCI.

How is context built in a UbiComp home environment? The use of mobile and wearable devices is paving the way for effectively building user context. It is likely that those devices are carried by users throughout the day and thus knowing the user's context.

As discussed earlier, there is a distinction between active and passive context aware systems. Too much of an active system and its users would probably feel a loss of control and as if decision making was in the hands of someone else. Imagine walking into your living room and things would start happening automatically – based on your context. To some users this would be fine, while others would be put off by it. It is all about finding the *sweet spot* – a system that makes use of context but waits for the users to implicitly perform an action before interpreting it and presenting relevant information to the user.

In the final conceptual design, users are located in a context aware living room - meaning that the living room has some knowledge about who the user is and what the user's intentions are. This knowledge is due to the user carrying personal mobile and wearable devices, in this case a smart phone and a smart watch/ smart wristband, that collect contextual data. The room is neither passive nor active in its context awareness, but something in between. It therefore waits for the user's implicit actions, before making any inferences on what the user wants to do. In this case, the implicit actions are a number of gestures.

Architectural Design As stated in Section 2.1 pg. 8, a UbiComp system is said to consist of:

- Smart Environments
- Smart Devices
- Smart Interactions

In the final design, the virtual living room is said to be a smart environment. In this living room there is a collection of interconnected CEs. All of the CEs in this environment are said to be smart devices and have the ability to send and receive data.

As seen in Figure 4.5, the virtual character has a personal smart device strapped to the right wrist. This device has been worn during a long period and thus collected the context of the character. This smart device is said to know what CE it is pointing towards, at any given time. The wrist worn device can also measure the state of the wearer's hand, i.e. is the hand open or closed.

In Figure 4.6, a tablet can be seen in front of the character. This tablet is said to be a shared smart device and its main purpose is to enable a portable platform for controlling all available devices. According to Brush et al [33] *"71% of the tablets were attributed to shared ownership"*, which should make for a good platform in a home environment for remotely controlling devices. However, control could also be held in a personal smart device (e.g. a smart phone). Other smart devices seen

Figure 4.5: Virtual character from behind

Figure 4.6: Virtual character from top

in Figure 4.5 and Figure 4.6 is the television set, which can show content sent to it, the game console that enables users to play games, the receiver that enables good sound from the speaker set and a printer. Moreover, there is a hidden gaze tracking system that enables users to do much of the same interactions enabled by the smart wristband.

All of these CEs and their connections provide the capabilities for "smart" interactions. In the final design, four "smart" interactions were developed and they are presented, discussed and motivated below in *Device Discovery* and *Device Selection*.

Final Conceptual Design

The four interaction concepts were designed so that they fulfil the requirements mentioned in Section 4.1 pg. 27.

Device Discovery The two device discovery methods designed, Gaze and Gesture, consists of two previously created concepts, *Augmented Descriptions* and *Natural Gestures* see AppendixA.1. According to Norman pg. 3 [28] there are two important notions to be aware of in order to make for good and consistent device discoverability.

> *"Discoverability: Is it possible to even figure out what actions are possible and where and how to perform them?"*
>
> *"Understanding: What does it all mean? How is the product supposed to be used? What do all the different controls and settings mean?"*

According to Norman pg. 10 [28] good discoverability also stems from

> *"Discoverability results from appropriate application of five fundamental psychological concepts...affordances, signifiers, constraints, mappings and feedback."*

In Kinect for Windows Human Interface Guidelines [34] a number of good guidelines are drawn up for interaction methods utilising gestures:

> *"Make sure your interactions are appropriate for the environment in which your application will be used." "Input methods should be reliable, consistent, and convenient..." "...we recommend that you keep the number of gestures small..."*

Gaze This concept allows the user to discover the identity and location of CEs in the user's vicinity by looking around. Discoverable devices are signified by the presence of a augmented signifier beside the device. When a discoverable device is in the centre of view for more than a certain amount of time, first and foremost a second signifier is presented as means of feedback. After a while, a descriptive card with information on the focused device is augmented next to the device as to solve issues with mapping. When the device is no longer in focus, the card disappears.

Gaze tracking enables *hands free interaction*, at the same time it enables users to use the *innate gesture* that they normally use when visually searching for entities.

The concept emphasises *ease of use* and *social acceptance*. It also strive to bridge the gaps between the *Gulf of Execution* and the *Gulf of Evaluation* pg. 38-40 [28].

In the setup there was no possibility for true eye gaze since there was no access to a real eye tracker. Instead the centre of the viewing field was used as a tracking point. When using the centre of the viewing field microsaccades do not need to be considered. Microsaccedes are otherwise a big issue [35].

Gesture The Gesture concept is similar to the Gaze concept, with the difference that searching for devices is done with a hand, performing a deictic gesture. Deictic gestures are gestures with the purpose of establishing the identity or location of an entity[36]. For the concept to work in reality there is a need for spatial location and rotation tracking of the arm. This could be done via a smart wrist band or similar wearable technology with ability for spatial location tracking.

The concept uses the descriptive cards that play a key role in the Gaze concept and it emphasises ease of use and at the same time a high perceived control.

Device Selection Two concepts for device selection were created: Grab to select and Push to select. These concepts aim to simplify the interaction between a user and the user's CEs. This is done by enabling natural interaction in a UbiComp environment. The two concepts are described and design choices are motivated below.

Grab to select Figure 4.7 shows the different steps (*a*, *b* and *c*) of Grab to select, implemented in the demoing environment.

This concept allows the user to select a device to interact with, by performing a grabbing gesture *(a)*. If a device was correctly selected, the user receives vibrational and visual (shifting colours) feedback through the smart watch/ smart wristband *(b)*. The device remains selected as long as the user keeps making a fist. The next step is to move the closed hand to the surface of a smart device, in this case a tablet, and open the closed fist *(c)*. If the fist is opened in mid-air, the grabbed device is no longer selected. By opening the hand on the tablet surface, the tablet renders a UI for media playback.

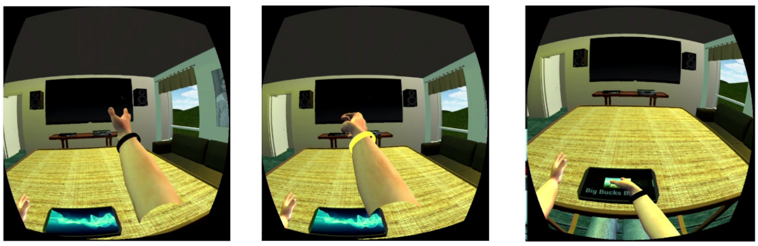

Figure 4.7: Grab to select: a, b and c

Grab to select is designed to give the user as much control as possible, by utilising the user's implicit actions in a UbiComp environment – offering interaction at the users fingertips. It is a *device-centred* concept, meaning one would first decide what device to interact with and secondly what is to be done with the device.

The fundamental principle of this concept is an analogy: grabbing hold to something means wanting to interact with it (e.g. grabbing a book). Another important principle in this concept is feedback, vibrational and visual. This feedback tells the user that the implicit action was interpreted by the environment as valid input. During the focus group (Section 4.2.2 pg. 32) the need for feedback when grabbing hold of something that is not physical, emerged. The vibration and colour shift is implemented to solve this issue.

The purpose of the analogy and the type of feedback mentioned above, is to make for a fairly well known mental model. This helps to bridge the *gulf of evaluation* ([28] pg. 38-40) – hopefully making the interaction feel natural and familiar, without a steep learning curve for the users.

Push to select Figure 4.8 shows the different steps (*a* and *b*) of Push to select, implemented in the demoing environment. This concept allows the user to first select media content and then select the output device by swiping the content towards it. The starting point of this concept is a smart device displaying the user's content *(a)*. While viewing the content, the user decides to flick the content to another device. The device that the user flicks towards is chosen and the content is transmitted to it *(b)*.

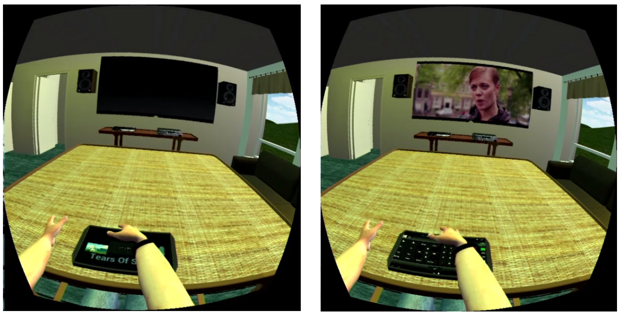

Figure 4.8: Push to select: a, b and c

Push to select is a *content-centered* way of interacting with ones CEs - meaning that the user first chooses what content to display and secondly what device to display it on. This concept is therefore best suited for interacting with devices that display, or in some other way, use content.

The founding stone of this concept is the familiarity of a swiping gesture. The familiarity depends on the *affordance* offered by the smart devices display - inviting users to perform swipe gestures on it. An affordance is defined as the relationship between the properties of an object and the capabilities of the agent that determine just how the object could possibly be used ([28] pg. 11-19).

Another important factor is natural mapping - taking advantage of spatial analogies that leads to immediate understanding ([28] pg. 20-22). For example, to move an object up, move the control up. The purpose of the familiarity mentioned above and natural mapping is to support the user in the execution of the interaction - hopefully making for natural interaction without a steep learning curve.

4.3 Implementation phase

This section presents and discusses the results of the Section 3.3 pg. 25. The implementation phase resulted in a hardware and software setup. Discussions adhering to the setup is in Section 4.3.1. Section 4.3.2 discusses the software parts of this setup.

4.3.1 Setup

Here the final hardware setup is presented, seen in Figure 4.9 pg.40 on the left hand side denoted as **P**.

1. Oculus Rift, a Virtual Reality Headset for 3D Gaming [37] - Used for head tracking and has three rotational degrees of freedom. This enables creating a virtual environment with high immersion.

2. Samsung Galaxy Nexus - This phone was attached to the user's wrist with two tufts. The phone was used to convey haptic feedback by means of vibrations. The phone also simulated the smart watch/ smart wristband worn by the character in the virtual living room, which in theory should lead to higher presence. Connection with the computer running the simulations was done through WiFi, which had a slight drawback due to the latency introduced - yet the benefits of not having a cable connection were great.

   Before settling upon using this device as haptic output device, experiments were performed with different communication methods and devices. None of the other methods reached the demands for low latency, stability and a cable free solution that was given via connecting a smart phone to the computer over Android Debug Bridge through WiFi. With this solution a connection is established to the phone from the computer, "root"[3] is gained and one can directly write to the vibrator via `/sys/class/timed_output/vibrator/enable` from within the simulator.

   The drawback of this was that the device used needed to be "rooted". For intended purposes this was a fast to implement solution with a high yield, since there were low overhead and the operation was sufficiently fast in comparison to using a real wearable device.

3. Razer Hydra Gaming Controller, PC Motion Sensing Controllers [17] - These hand controllers utilise magnetic tracking. They have high precision and the controllers are able to track in six degrees of freedom.

   The controllers are attached to a pair of working gloves[4] with Velcro. This makes for easy set-up with the user and enables the user to be hands free.

[3]On a "rooted" device one can gain "root", which from a Unix standpoint is being system administrator with full read and write access to all parts of the system.

[4]A major drawback was the fact that only one pair of gloves was used. This lead to the controllers not being firmly attached should the test subject have smaller hands. This in turn could lead to bad tracking

With the controllers the user's hand movements could be mapped to the virtual character, to make for natural movement inside the virtual environment.

4. 5DT Data Glove 5 Utra [32] - These are a pair of finger movement tracking gloves. They are usually used for motion capturing and animation but since they can be used in real time, they were mapped onto the fingers of the virtual character.

5. Sony Tablet Z - This tablet was used for capturing gestures such as swipes, taps and long presses. The input on this device was translated into the virtual environment. Since there is a tablet in the virtual environment at "the same place" as the physical tablet, this also would lead to higher presence since users can get the tactile feedback from the physical tablets glass at the same time as they saw things happen in the virtual environment.

   The solution used was simple application that was able to take Debug Logs of touch events on the tablet was implemented. The touch events were read from the device via Android Debug Bridge, over a USB cable, from within the demoing environment. This solution meant that there was full interchangeability of the tablet.

6. Razer Hydra Gaming Controller - This is the base station of the input device. It emits the magnetic field that is induced in the hand controllers.

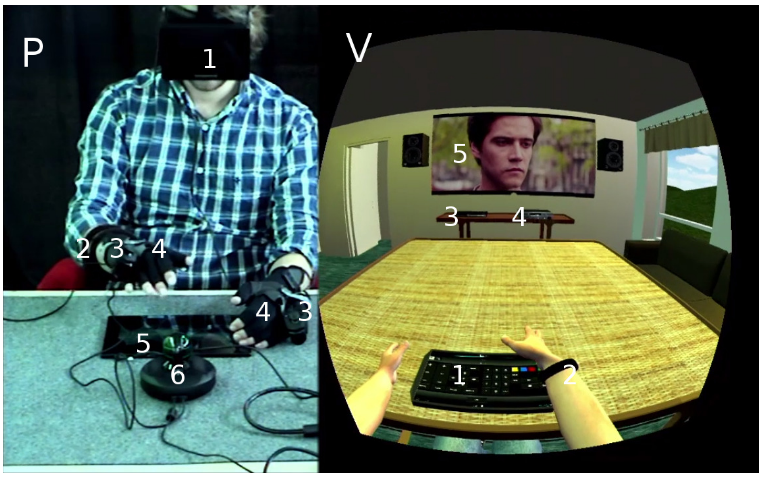

Figure 4.9: The physical world (P) vs. the virtual world (V)

Additionally to the setup seen in Figure 4.9 pg. 40 a HP Z400 workstation was used. This was the computer used to run the software and the virtual environment. Specification wise the computer was fitted with a dual core Intel Xeon Processor as well as 12 GB ram and most importantly an Asus 680 GTX graphics card[38]. The high end graphics card meant that the rendered video feed to the

Oculus Rift could hold a stable frame rate. The graphics card also enabled multiple monitors so that a cloned view of what a user saw in the Oculus Rift could be viewed externally.

The virtual environment corresponding to the physical world is denoted **V** on right hand side of Figure 4.9.

1. The virtual tablet - This is where users see themselves doing tablet inputs
2. Smart watch/smart wristband - This virtual device gives users visual feedback at the same time as the smart phone in the physical world gives haptic feedback.
3. Virtual Game Console - A game console, for experiencing gaming and multimedia
4. Virtual Receiver - A receiver is where audio devices are interconnected with the speaker set
5. Virtual TV - A virtual television set.

4.3.2 Software

In this section there will be a rundown of the different software components, both third party and our own solutions, that made the creation of the demoing environment possible.

Android Debug Bridge Via Android Debug Bridge[39] (adb) one can gain information and transfer information to and from the device. Should the connected device be "rooted", a user can also perform system operations on the device.

AutoDesk 3ds Max 3dsMax[40] is a software solution for animation and 3d modelling. During the course of this thesis many models were created with this software suite. Some models that where gotten on the asset site TurboSquid [41] where firstly imported here and manipulated to fit our purposes.

Gestures.apk This is the program created to enable logging of touch input on the tablet. Touch events are logged and later on read from the logcat [42] part of adb. It consists of a simple view, which only shows a picture. This enables a subject to quickly see if the program is running or if the screen is off.

GIMP Image Editor GIMP[43] is the free software that was used to create and edit textures and images used throughout this thesis.

FFmpeg In order to use video clips as a texture, Unity needs the clips to be in the format of `*.ogv`. The conversion can be done inside Unity by installing Apple QuickTime. This was not a good solution in a windows environment, thus FFmpeg[44] was installed and used. FFmpeg has enormous support for a multitude of formats and the user has full control of the output file.

Inverse Kinematics by Dogzer This collection of scripts is available in the Unity Asset store for free[45]. The inverse kinematics script was used to enable the input from the Razer Hydra to control the arm limbs of the virtual character. Without inverse kinematics the movement of the arms would look bad, and writing the inverse kinematics would be time consuming.

LeanTween LeanTween[46] is a lightweight tween engine used to animate some objects in the demoing environment.

Oculus SDK The Oculus SDK[29] comes with added functionality to the Unity game engine. This helps developers to build scenes with the Oculus VR Headset as intended target.

Project Pinocchio This free to use web application[47] is a project by Autodesk that enables fast creation of 3D characters. It also supplies the created character with a rigged skeleton for natural looking animations and limb control.

Sixense Unity Plugin This plugin[48] is packaged in a unitypackage, available in the Unity asset store. When imported into the Unity Game Engine, the developer can track the translation of the controllers and use this as an input.

Unity Game Engine The Unity Game Engine[19] is mainly used to develop cross-platform games. There is a version available for free and a Pro licence for more functionality. Since external libraries are used, a Pro version is needed to create and build scenes. Unity3D gives developers choices of implementing in C#, javascript and Boo. C# was chosen to implement the concepts for the demoing environment.

4.4 Evaluation

Through literature research important topics for usability tests were concluded. Also, thorough discussions with the advisors lead to the creation of hypotheses which are presented in Section 4.4.1 pg. 44. This section will thoroughly discuss the test setup decided upon in Section 3.4 pg. 25.

4.4.1 Usability Test

To evaluate and compare the interaction concepts, a usability test was performed. Here the test plan, the type of data to collected during testing and the hypothesis for the usability test is defined.

Before the test started, the test subject would be brought up to speed with the concept of UbiComp. The background information can be read in Appendix A.2 pg. 82.

The usability test consisted of testing the four interaction concepts on test subjects, one concept at a time. The test subject were given tasks which they were asked to perform. Each category had its task:

- For the *Device Discovery* the task was to find the device corresponding to video playback.
- For the *Device Selection* the task was to start video playback on that device.

After a task was finished, the test subject would fill out a NASA TLX questionnaire [49], to be able to measure the perceived workload for the specific interaction.

When the questionnaire had been filled out, there was a structured discussion, asking the test subject a couple of questions. When the subject had answered the questions, there would be an unstructured discussion about their experience. Each test had a set of specific questions. The questions can be found in Appendix A.3 pg. 84.

Since there were two *Device Discovery* and two *Device Selection* interaction concepts, comparative questions would be asked after they had tried both interaction concepts in one category. When all tests were done, a number of questions were asked regarding how the test subject felt about a ubiquitous future, and their general thoughts about logging context information.

What is being tested?

The aim for the usability test was to find out whether the interaction concepts for a UbiComp home environment fulfil the requirements set up in Section 1.2 at pg. 3.

Who are the test subjects?

The process of getting test subjects consisted of sending out invitations to well over 500 students in Lund. During the pilot testing, it was established that the test-pool should consist of at least 20 people, with as wide distribution in age as possible. Also, a equal amount of men and women would be preferred.

Test Setup

The test was conducted in the usability lab at IKDC[5], Lund. This usability lab is a controlled environment, with audio and video capturing capabilities. All tests were captured with audio and video for the purpose of analysis. The setup of the testing environment can be seen at Figure 4.9 at pg. 40.

Data Collection

During the test both quantitative and qualitative data was collected.

[5]Ingvar Kamprad Design Center - the home of Design Sciences at the Faculty of Engineering, Lund University

Quantitative Data The quantitative data collected was going to be used in statistical analysis. The following data was collected:

- Time to finish the task at hand
- Number of errors
- Cost of errors in time
- NASA TLX - the perceived workload - a questionnaire with six categories, measuring on a scale $1 - 20$:
  - Mental demand - How mentally demanding was the task?
  - Physical demand - How physically demanding was the task?
  - Temporal demand - How hurried or rushed was the pace of the task?
  - Performance - How successful were you in accomplishing what you were asked to do?
  - Effort - How hard did you have to work to accomplish your level of performance?
  - Frustration - How insecure, discouraged, irritated, stressed, and annoyed were you?

To be able to calculate the perceived workload, the six NASA TLX categories are weighted against each other. How this was done can be seen in Appendix A.4 pg. 85.

Qualitative Data Qualitative data was collected, by transcribing the test subjects answers to questions. This data was to some degree turned into quantitative data by reading out tendencies in preferences (the test subjects were asked which interaction concept they preferred - they were also asked questions about their thoughts on context logging).

Hypotheses The following hypothesis were to be tested for in the usability test.

$H1$ Age should not significantly affect the time to finish the task.

$H2$ Gender will not significantly affect the time to finish the task.

$H3$ For any of the discovery and selection methods, there will be no significant correlation between the time to finish a task and the perceived workload.

$H4$ There will be no significant relation between perceived frustration and what interaction concept is preferred.

$H5$ There will be no significant difference in time between the two *Device Discovery* methods, Gaze and Gesture.

$H6$ There will be no significant difference in time between the two *Device Selection* methods, Grab and Push.

H7 There will be no significant difference in the physical demand for the two *Device Discovery* methods, Gaze and Gesture.

H8 There will be no significant difference in the physical demand for the two *Device Selection* methods, Grab and Push.

Pilot testing

Before conducting a usability test, it is good practice to perform pilot testing. In pilot testing one would perform a dry run of the test setup as well as all test protocols. This is made in order to find bottlenecks in the usability test. Performing pilot testing also makes the test leader more confident in how to perform the test and it gives the ability to create a set of frequently asked questions. From pilot testing one can draw conclusions of how large the test-pool needs to be in order to gain valid statistics. After the pilot testing, a last iteration of the implementation was undertaken. The test protocols set up for the pilot testing were revised, in order to be used in a usability test.

4.5 Comparison to other work

Several methods and tools have been used for prototyping ubiquitous interaction. For example Carter and Mankoff [50] used paper prototypes to identify useful aspects of their system design and found it useful for rapid iteration of graphical design. However, their approach offered low fidelity resulting in limitations, such as static user interfaces.

Another well-known prototyping method widely used within HCI and UbiComp is WOZ. The advantages with a WOZ setup include flexibility, mobility and the ability to combine different devices. ConWIZ [51] is a WOZ tool which includes a mobile application that is capable of controlling the simulation of a WOZ prototype as well as contextual objects such as fans, lights etc.

One example mentioned earlier is the VIEW system developed by NASA Ames Research Center [15]. The system has similarities with the VR setup presented in this thesis since it uses a HMD and tracking of upper extremities.

Courter et al. [52] developed a highly immersive and sophisticated interactive virtual environment for designing space stations. The system uses a CAVE^{TM} [53] like setup with an omnidirectional treadmill thus making it a good example of a high fidelity and high cost setup.

In the prototyping methodology presented in this report, the approach is more interactive compared to the one used by Carter and Mankoff thus offering a more dynamic experience. WOZ setups, like the ConWIZ, introduce human factors, such as the inability of the wizard to keep up with the test subject, which can influence usability testing. This a non issue with a setup like the one proposed in this report, since the interaction performed is not influenced by the test leader. As mentioned earlier, the VIEW setup has similarities to the proposed setup. However, VIEW does not use off-the-shelf devices and can thus be considered expensive. The virtual environment created by Courter et al. offers high fidelity at a high cost. One could argue that the setup proposed also offers high fidelity but at a fraction of the cost, while being more mobile and flexible.

Chapter 5

Usability Test: Results & Discussion

The interaction concepts were implemented and usability testing was performed to evaluate them. The results from the usability test and discussions are grouped hypothesis wise in Section 5.1. All calculations were done using **R** [54] and all generated plots were created with **ggplot2** [55]. Furthermore, the qualitative data and a general discussion on the interaction concepts is presented in Section 5.1.2 and Section 5.1.3 respectively. Finally, possible sources of error, Section 5.2 and a discussion on the demoing environment, Section 5.3, are presented.

5.1 Results and Discussion

Since the audio fell out in some of the usability tests and preferences were read out by transcribing the qualitative data, hypotheses concerning preferences were only tested on a subset of 20 of the test persons instead of 24, which was the full test-pool.

To gain insight on how the different interaction concepts are related to one and another, a plot was done of their mean NASA TLX values. This plot can be seen in Figure 5.1. Also, in Table 5.1 age and gender distribution, is presented.

Table 5.1: Information on test subjects

Age	19-24	25-30	31-36	37-42
N	17	3	3	1
Female	7	2	0	0
Male	10	1	3	1

5.1.1 Quantitative Data

The results of the correlation tests performed for hypotheses $H1$–$H3$ are presented in this section. Also presented are the results from the Welch t-tests, performed for hypotheses $H4$–$H8$.

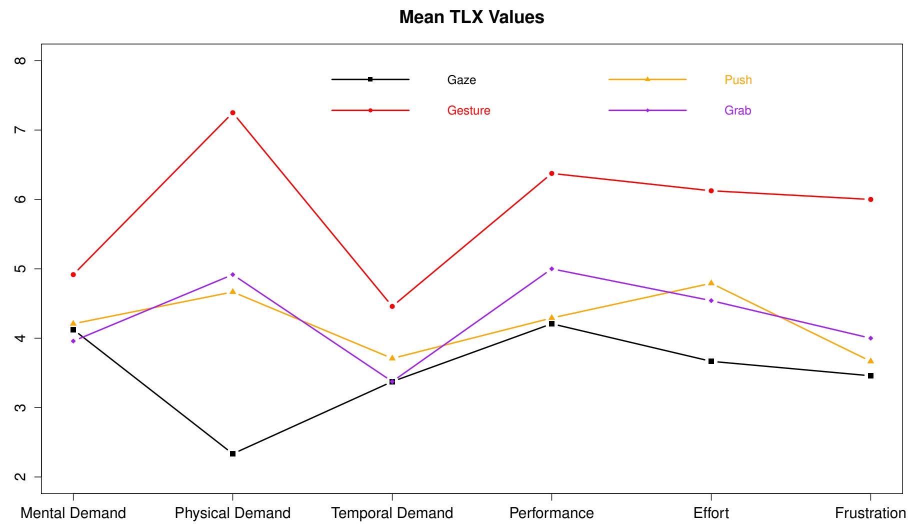

Figure 5.1: Mean NASA TLX values for all interaction concepts

For each of the hypotheses *H*1–*H*3 a table is presented with relevant data. The number of participants is $N = 24$ and the *p-value* is the significance level. *r* is the correlation coefficient, where a value near 0 means no correlation.

*H*1 - Age should not significantly affect the time to finish the task

Correlation between age and the time to finish the task is represented by Table 5.2.

Table 5.2: Correlation between Age and Time

Category	Test case	p-value	r
Device Discovery	Gaze	0.9875	-0.003370612
	Gesture	0.7274	-0.07505965
Device Selection	Grab	0.2219	0.2589014
	Push	0.2381	0.2503004

As can be seen in the data presented above, no significant correlation was found, between age and time to perform any of the task given for each *Discovery Method* and *Selection Method*. This hypothesis could therefore not be rejected.

A problem with this test was that the age distribution in the test pool was poor. If the age distribution and test pool had been larger, the result may had been different. The age distribution is visualised in Table 5.3.

Table 5.3: Age of Test Subjects

Age	19-24	25-30	31-36	37-42
N	17	3	3	1
%	70.83	12.5	12.5	4.17

*H*2 - Gender will not significantly affect the time to finish the task

Correlation between gender and the time to finish the task is represented by Table 5.4.

Table 5.4: Correlation between Gender and Time

Category	Test case	p-value	r
Device Discovery	Gaze	0.06262	-0.3857789
	Gesture	0.5853	0.1172704
Device Selection	Grab	0.9121	0.02379498
	Push	0.6452	0.09904081

As can be seen in the data presented above, no significant correlation was found, between gender and time to perform any of the tasks. This hypothesis could therefore not be rejected.

*H*3 - For any of the discovery and selection methods, there will be no significant correlation between the time to finish a task and the perceived workload

Correlation between the time to finish the task and the perceived workload is represented by Table 5.5.

Table 5.5: Correlation between Time to finish and workload

Category	Test case	p-value	r
Device Discovery	Gaze	0.02011	0.4712238
	Gesture	0.00869	0.5232866
Device Selection	Grab	0.5826	0.1180932
	Push	0.3483	0.2001994

As seen above, significant correlation was found in both of the *Device Discovery* test cases. Testing on the two *Device Selection* methods found no significant correlation. The mean perceived workload for Push is ≈ 4.44 and ≈ 4.47 for Grab. Thus this hypothesis can be rejected for the *Device Discovery* methods. However, this hypothesis could not be rejected for the *Device Selection* methods.

For *Device Discovery* it is shown that the higher perceived workload, the longer time the task will take.

For the test case Gaze, the mean perceived workload was ≈ 3.34 and mean perceived workload for Gesture was ≈ 6.05. Gesture is therefore perceived as more demanding. This could be traced to the fact that in Gesture, a user performs a deictic gesture while also using the eyes to find the device. Gesture is therefore perceived as having an extra step. This was also seen in the qualitative data. Two comments given during the test sessions were:

"Extra step in gesture.." - Comparing to Gaze
"Using eyes was easier" - Comparing to Gesture

The higher perceived workload can be explained by considering the mean physical demand, one of the six categories in NASA TLX. The mean perceived physical demand for Gaze is ≈ 2.33 and 7.25 for Gesture. This would be in line with the two comments mentioned above and could therefore explain why gesture has a higher perceived workload.

For each of the hypotheses $H4$–$H8$ a table is presented with relevant data. The *p-value* is the significance level. Also, the mean value is presented. The number of participants N is specified in the table caption for each test, since the number of participants is different due to malfunctioning audio recording.

$H4$ - There will be no significant relation between perceived frustration and what interaction concept is preferred

Tables for this hypothesis is Table 5.6 and 5.7. They represent Welch t-testing between NASA TLX frustration and which interaction method was preferred. The test case column shows what the current test case was. If a subject preferred a certain discovery method the subject is grouped in that column.

Table 5.6: Comparison between mean NASA TLX Frustration for Device Discovery grouped by preference. N=20

Category	Test case	mean prefer Gaze	mean prefer Gesture	p-value
Device Discovery	Gaze	2.83	3.44	0.5901
	Gesture	5.83	5.0	0.6913

Table 5.7: Comparison between mean NASA TLX Frustration for Device Selection grouped by preference. N=20

Category	Test	mean prefer Grab	mean prefer Push	p-value
Device Selection	Grab	4.00	3.33	0.707
	Push	4.22	2.67	0.1712

As can be seen above no significant relation was found. This hypothesis could therefore not be rejected.

This could simply be due to that preference of an interaction method depends on so much more than only frustration. A parameter that cannot really be tested with the quantitative data at hand, but which qualitative data supports, is the fact that people like what they are used to:

"Push is similar to the Plex in my home." - Prefers Push
"It felt natural, even today I use my eyes to discover devices" - Prefers Gaze

Preference of a interaction method could also be related to the feeling of being in control. Examples of this is:

"..you can be more specific, I'm more in control.." - Prefers Gesture
"Gesture gives more control.." - Prefers Gesture

To conclude, preferring one of the *Device Discovery* methods or *Device Selection* methods over the other is not related to the perceived frustration. In the raw quantitative data it was shown that 12 out of 20 test subjects preferred Gaze over Gesture. Also, 12 out of 20 preferred Push over Grab.

H5 - There will be no significant difference in time between the two *Device Discovery* methods, Gaze and Gesture

The table and figure for this hypothesis is Table 5.8 and Figure 5.2, and presented data represents Welch t-testing between the time to finish the task (measured in seconds) and the two different discovery methods. The figure shows the distribution in time for the different discovery methods.

Table 5.8: Comparison in time between Gaze and Gesture. N=24

Category	mean time Gaze	mean time Gesture	p-value
Device Discovery	30.58	39.58	0.3004

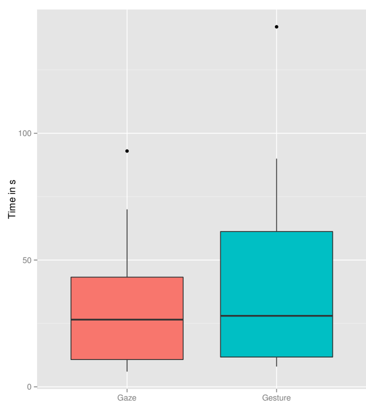

Figure 5.2: Gaze vs. Gesture, time

As can be seen above no significant relation was found. This hypothesis could therefore not be rejected.

*H*6 - There will be no significant difference in time between the two *Device Selection* methods, Grab and Push

The table and figure for this hypothesis is Table 5.9 and Figure 5.3. The table represents Welch t-tests between the time to finish the task (measured in seconds) and the two different selection methods. The figure shows the distribution in time of the different selection methods.

Table 5.9: Comparison in time between Push and Grab. N=24

Category	mean time Push	mean time Grab	p-value
Device Selection	13.54	28.50	0.001968

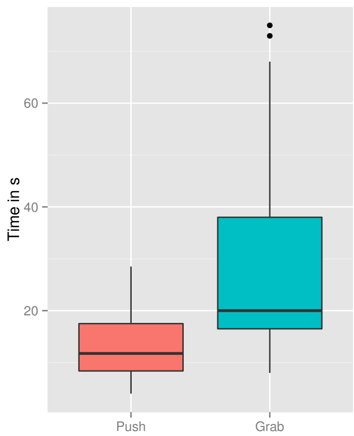

Figure 5.3: Push vs. Grab, time in seconds

As can be seen in the data presented above a significant difference was found and the mean time to finish for Grab was 28.50 seconds and 13.54 seconds for Push. Thus the hypothesis is rejected.

The difference in time might be explained by the following comments from the qualitative data:

"..time consuming with the extra step in Grab.. "
"..it's more convenient to push content towards a device."

To further investigate this result, two additional correlation test between the time to finish the task and the perceived NASA TLX physical demand were done. For Gaze the *p-value* = *0.67* - meaning no significant correlation was found. For Gesture the *p-value* $\approx$ *0.02* - meaning there was a significant correlation. This could explain the two comments, since the extra step in Gesture could be perceived as being more physical demanding, hence more time consuming.

H7 - There will be no significant difference in the physical demand for the two *Device Discovery* methods, Gaze and Gesture

Table for this hypothesis is Table 5.10, and corresponding figure is Figure 5.4. The table represents Welch t-tests in between the two discovery methods and their mean NASA TLX perceived physical demand. The figure shows the distribution in physical demand for the different discovery methods.

Table 5.10: Comparison in Physical demand between Gaze and Gesture. N=24

Category	mean demand Gaze	mean demand Gesture	p-value
Device Selection	2.33	7.25	8.931e-05

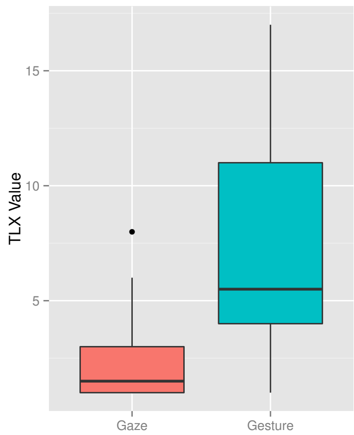

Figure 5.4: Gaze vs. Gesture, NASA TLX physical demand

As can be seen in the data presented for this hypothesis a significant difference was found. Thus this hypothesis can be rejected.

The result is not surprising, since in Gesture a deictic gesture is used, whilst in Gaze only the eyes are used to visually discover devices. The mean NASA TLX physical demand was for Gaze $\approx$ 2.33 and for Gesture 7.25.

H8 - There will be no significant difference in the physical demand for the two *Device Selection* methods, Grab and Push

Table for this hypothesis is Table 5.11, and corresponding figure is Figure 5.5. The table represents Welch t-tests in between the two selection methods and their mean NASA TLX perceived physical demand. The figure shows the distribution in physical demand for the different selection methods.

Table 5.11: Comparison in Physical demand between Push and Grab. N=24

Category	mean demand Push	mean demand Grab	p-value
Device Selection	4.67	4.92	0.7815

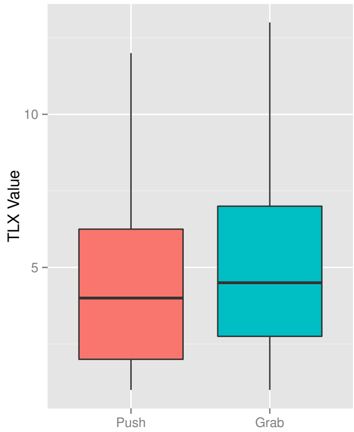

Figure 5.5: Push vs. Grab, NASA TLX physical demand

As can be seen in the data presented above no significant difference in physical demand was found. This hypothesis could therefore not be rejected.

5.1.2 Qualitative Data

Here a selection of additional comments given during the usability testing. Both positive and negative comments are presented. These comments were chosen to give a fair overview of the test subjects general thoughts.

Gaze

Positive:

"It was very natural"

"It felt intuitive"

Negative:

"When the cards kept appearing I was overwhelmed"

" ... can be annoying with too much information, all the time popping up"

Gesture

Positive:

"... is comparable to pointing towards objects one is interested in ..."

"... you feel more comfortable/secure you can be more specific. I'm more in control."

Negative:

"... cool in movies, but if it was my living room I wouldn't want to move to much... you know I'm a lazy programmer so, the hand moving is a lot to ask"

"Feels weird, not a fan"

Push to select

Positive:

"... a very convenient way to select and start ... "

"I think it's comfortable ... its intuitive to send things in the right direction"

Negative:

"I think its more logical to start with the device"

"... when having many devices I wouldn't have felt comfortable to send things like this"

Grab to select

Positive:

> *"Cool idea and it felt very intuitive"*

> *"... if you just want to watch TV then just grab"*

Negative:

> *"It felt somewhat unnatural ... a bit weird. When grabbing towards something, you want to feel what you've grabbed in the hand, here you grab the thin air."*

> *"This is not as easy as push"*

5.1.3 General discussion on interaction concepts

Based on the result, some insights might be gained - as to if the interaction concepts fulfil the earlier stated requirements.

Device discovery in a UbiComp environment In a UbiComp home environment one can argue that there is low necessity for device discovery, since all devices are interconnected and the person that wishes to interact with a service in most cases know what device supports this. However, there are some use cases when device discovery could be beneficial and why should this device discovery be troublesome. For instance if you have a guest over and they wish to listen to the radio. A simple device discovery method would aid them in performing this task.

Device discovery methods could also be beneficial in public environments. During the usability test, the test subjects proposed a lot of different situations where the different discovery methods could be used: finding the milk in a convenient store, finding the ticket vending machine in a parking garage or at a train station as well as finding deals in shops one is passing by etc.

Gaze By observing the discussion for hypothesis *H*3 as well as hypothesis *H*4 and *H*7 - together with the qualitative data presented for Gaze, one could say that: *Gaze yields a low workload, it requires low physical demand, it is familiar and natural.*

The above mentioned might make for simpler device discovery. However no such direct conclusions will be made, since there is no clear definition of what simple device discovery is. It is however found that this type of interaction yields in a low workload, mainly due to the fact that it is familiar, natural and requires low physical demand. Therefore, it might be concluded that this kind of interaction could be acceptable to some users for some use cases.

Gesture By observing the discussion for hypothesis *H*3 as well as hypothesis *H*4 and *H*7 - together with the qualitative for Gesture, one could say that: *Gesture is moderately-to-low demanding, it is moderately-to-low physically demanding but familiar and gives the user the feeling of being in control.*

It if left unsaid whether the above mentioned makes for simpler device discovery. Once again, there is no clear definition of what simple device discovery is and therefore no direct conclusion will be made. However, since this kind of interaction was found to be moderately-to-low demanding, while offering a high level of control and being familiar, it might be concluded that it could be acceptable to some users for some use cases.

Comparing the two device discovery methods When comparing Gaze and Gesture one can start to look at the time the different tasks took. As presented in hypothesis *H*5, no significant difference between the two interaction methods was found. Furthermore, in hypothesis *H*7, a significance for Gaze being less physical demanding than gesture was found. The perceived workload was found significantly lower for Gaze compared to Gesture. When reviewing the qualitative data, the comments given during the usability test, one could see that users feel as if they had more control when using Gesture - while Gaze felt more natural. In the discussion for hypothesis *H*4, one can see that a similar amount of test subjects preferred either of the methods. If anything else, this goes to show that simple device discovery is a subjective assessment and that the both discovery methods could be good to use, depending on users and use case.

Device selection in a home environment During the bodystorming sessions (Section 4.1 pg. 27), a number of points were made when investigating how interaction works today. By introducing implicit HCI (by the means of gestures) and minimising the amount of controllers and other interaction methods, the two device selection methods Grab and Push aim to challenge today's interaction paradigm - hopefully making interaction with CEs feel simpler and more intuitive.

Grab By observing the discussion adhering to hypothesis *H*3 as well as hypothesis *H*6 and *H*8 - together with the qualitative data presented for Grab, one could say that: *Grab to select yields a low workload, is suited for quick selection of devices and requires low physical demand.*

The above mentioned might make for simpler device selection, but no such conclusion will be made - since there is no clear definition of what simple device selection implies. It was however found that this type of interaction yields in a low workload for users, could be quick to use for some use cases and requires low physical demand. For some users it did not feel natural, since they are grabbing in thin air, while other felt that it was intuitive. It might be concluded that this kind of interaction could be acceptable to some users for some use cases.

Push By observing the discussion adhering to the testing of hypothesis *H*3 as well as hypothesis *H*6 and *H*8 - together with the qualitative data presented for Push, one could say that: *Push to select yields a low workload, requires low physical demand, is convenient and feels intuitive.*

Whether the above mentioned makes for simpler device selection is left unsaid, since no definition of simple device selection exists. It was however found that this interaction method yields a low workload for users, felt comfortable and intuitive as well as familiar (as one test subject put it: *"It is somewhat like pushing a DVD into a DVD-player"*). It might be concluded that this kind of interaction could be acceptable to some users for some use cases.

Comparing the two device selection methods When comparing Grab to select and Push to select one can start to look at the time the different tasks took. As presented in hypothesis *H*6, a significance for Push being quicker to perform than Grab was found. Not surprising, since Grab was percieved as having an "extra step". In the discussion for hypothesis *H*3 one can see that both methods yielded a similar and low workload. Furthermore, in the discussion for hypothesis *H*4, one can see that a similar amount of test subjects preferred either of the methods. If anything else, this goes to show that simple device selection is a subjective assessment and that the both selection methods could be good to use, depending on users and use case. In Figure 5.1 pg. 48, it can clearly be seen that the two methods for device selection are very much similar when compared in the different categories of the NASA TLX.

5.2 Possible sources of error

In this section, some possible sources of errors that could have influence the usability test are presented.

Gloves The gloves that were used to strap the user onto the Razer Hydra were only available in one size. This made for some issues for users with smaller hands. The problem was that, since the Hydra was not strapped as well as it should be, the tracking of the users' hands was lacking. This problem was relevant in the cases of testing Gesture, Grab and to some extent when testing Push.

This could have been alleviated by using multiple gloves to strap the Razer Hydra to. Since velcro was used as the strapping mechanism it would not take long time to try out a pair of gloves as to find the "perfect fit".

Cables All of the cables used in order to hook the users to the virtual environment[1] made for some problems with restricted movement. The problems became more notable when the test subject already had issues with the strapping mechanism to the Razer Hydra - since test subjects who already had problems with the size of the gloves were the ones that were the most effected. Had multiple

[1]The cables used for the gloves and the hydra were grouped together on the arm. The cable for the Oculus Rift went behind the head and downwards.

gloves been used much of the problems experienced by the cables would have been alleviated.

5.3 Discussion on demoing environment

The ability to gain results, those presented in this Chapter, shows that the demoing environment created during this master thesis is sufficient when visualising and testing complex interaction concepts. All test subjects finished the tasks given, and the mean NASA TLX performance was (on a scale $1-20$):

- Gaze ≈ 4.21
- Gesture ≈ 6.37
- Push ≈ 2.30
- Grab $= 5.00$

This goes to show that the test subjects felt as they succeeded well when performing the interactions - while being novel to the interaction concepts proposed and to the demoing environment.

As presented previously, there were two major sources of errors. These two errors might have influenced the result. However, looking at the *Device Discovery* concepts design, the level of inherited physical demand should be greater for Gesture. From the data in hypothesis $H3$ one can see that the perceived physical demand for Gesture is indeed greater then Gaze - showing that relations in the physical world have been translated to the demoing environment. With this said, one might say that large influences of different error sources have been avoided.

There are some critics of natural user interfaces. A notable article by Norman [56] describes some problems that can arise when technology is adopted before it is mature. Using a demo environment like the one presented in this master thesis enables maturing of these interaction methods, since one is given the possibility to test design choices at an early stage, in a quick and reproducible manner.

Without using a developing environment such as the Unity game engine, much of the interaction concepts developed during this master thesis would be very hard to visualise. This goes to show that a game engine can be used for so much more than implementing games. In fact, it proved to be crucial in fast prototyping of interaction concepts and offered the ability to quickly accept or reject an interaction concept.

An aspect of using a game engine as developing environment was the fact that some requirements for how the interactions should be performed could be drawn up, simply by implementing a quick Hi-Fi prototype, to see how it was experienced. As an example can be said that deciding the time it should take for a descriptive card to appear, when the eyes rested on a device, was decided by implementing a placeholder for a descriptive card. Later on, sitting in the demoing environment, different variables for how long a user needed to hold their gaze could be tested out, in order to find a time span that was neither too long - making for a poor and slow discovery method - nor too short - making for an obtrusive interaction method.The fact that a game engine makes for a quick, reproducible experiences with minor changes makes it perfect for prototyping and evaluation. Also, in conjunction with the demoing tools given during the start up of this master thesis, an immersive experience could easily be created.

Chapter 6

Conclusions

Based on the discussion in Chapter 5, the following conclusions can be drawn:

Gaze
yields a low workload, it requires low physical demand, it is familiar and natural.

Gesture
is moderately-to-low demanding, it is moderately-to-low physically demanding but familiar and gives the user the feeling of being in control.

Grab to select
yields a low workload, is suited for quick selection of devices and requires low physical demand.

Push to select
yields a low workload, requires low physical demand, is convenient and feels intuitive.

The VR methodology developed during this master thesis enables prototyping, showcasing and testing of UbiComp interaction concepts. The usability test and the analysis of the collected data, showed that the methodology offers good mediating capabilities for complex interaction concepts. At the same time, the methodology offers development in a fast, iterative and relatively cheap manner, while offering high fidelity.

Chapter 7

Future work

To further show the benefits of using the proposed methodology, more controlled experiments should be performed. Most importantly, experiments with a control group are needed in order to learn more about the validity of this methodology.

Furthermore, the VR setup has improvement potential in terms of hardware. For example, position tracking could add to the VR system's immersion. This could be solved with an external system, e.g. a Kinect camera, but the best strategy is probably to wait for the next generation of VR HMD's, such as the upcoming Oculus Rift Development Kit 2 [57]. A problem that was observed with the test setup, that may have influenced the data, was the fact that some test subjects had hand sizes a couple of sizes below the gloves used in the setup. As a consequence, hand gestures were hard to perform, which in the long run lead to a longer test time with more errors. A more flexible test set up, with different sized gloves, is needed.

Another possibility when using a game engine to develop prototypes would be to use see-through AR. This could enable prototyping person-to-person interaction as well as multiple user-CEs interaction for UbiComp environments.

Chapter 8

Relevance for industry

Prototyping complex or somewhat complex interaction with the methods and tools at hand today is inherently difficult and expensive, meaning that interaction designers and developers never get the chance to visualise their thoughts and test their designs at an early stage. This is hindering the further development of interaction itself. Another issue is that, early in a design process, it is often unclear if proposed interactions are acceptable to potential users. The ability for early acceptance or rejection of a design concept, based on usability testing is somewhat missing today. Further development of prototyping methods and tools aiding in prototyping is important in order to create new and interesting interaction. Using a methodology like the one presented in this report, corporations will get the ability to cost efficiently try out new designs, interactions and products.

When could you consider using VR prototyping? When you are developing wearable technologies that are to be used in a complex manner in novel ways or if you are developing novel interaction concepts, you could consider using VR prototyping. VR prototyping could also be good for testing out novel interactions, with novel sensor technology or for simply visualising conceptual ideas. It is probably best used in the early phases of a design process, when design choices need to be tested early on and often.

References

[1] R. T. Azuma, “A survey of augmented reality,” vol. 6, no. 4, p. 355–385. [Online]. Available: http://www.mitpressjournals.org/userimages/ContentEditor/1332945956500/PRES_6-4_Azuma_web.pdf

[2] M. Weiser and J. S. Brown. Bits flowing through the wires... [Online]. Available: http://www.ubiq.com/hypertext/weiser/acmfuture2endnote.htm

[3] The computer for the 21st century. [Online]. Available: http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html

[4] S. Poslad, *Ubiquitous computing: smart devices, environments and interactions*. Wiley.

[5] M. H. Coen, “Design principles for intelligent environments,” in *AAAI/IAAI*, p. 547–554. [Online]. Available: http://www.aaai.org/Papers/AAAI/1998/AAAI98-077.pdf

[6] A. K. Dey, “Providing architectural support for building context-aware applications.” [Online]. Available: http://www.researchgate.net/publication/27525468_Providing_architectural_support_for_building_context-aware_applications/file/79e415136b7245d03a.pdf

[7] ——, “Understanding and using context,” vol. 5, no. 1, pp. 4–7. [Online]. Available: http://link.springer.com/10.1007/s007790170019

[8] B. Buxton, “Designing for ubiquitous computing.” [Online]. Available: http://research.microsoft.com/apps/video/dl.aspx?id=188797

[9] ——, “Ubiquitous computing.” [Online]. Available: http://research.microsoft.com/en-us/um/redmond/events/techfest2013/bill-buxton-ubiquitous-computing-march-2013.pdf

[10] A. Schmidt, “Implicit human computer interaction through context,” vol. 4, no. 2, pp. 191–199. [Online]. Available: http://link.springer.com/10.1007/BF01324126

[11] ——, "Context-aware computing: Context-awareness, context-aware user interfaces, and implicit interaction," in *The Encyclopedia of Human-Computer Interaction, 2nd Ed.* [Online]. Available: http://www.interaction-design.org/encyclopedia/context-aware_computing.html

[12] M. Lombard and T. Ditton, "At the heart of it all: The concept of presence," vol. 3, no. 2, p. 0–0. [Online]. Available: http://dx.doi.org/10.1111/j.1083-6101.1997.tb00072.x

[13] D. W. Schloerb, "A quantitative measure of telepresence," vol. 4, no. 1, pp. 64–80.

[14] M. W. J. E. G. Alce, "Virtual reality i teori och praktik (MAM101)." [Online]. Available: http://www.eat.lth.se/fileadmin/eat/MAM101/VR-Kompendium2013.pdf

[15] S. Fisher, "Virtual environments, personal simulation and telepresence." [Online]. Available: http://itofisher.com/sfisher/portfolio/files/images/7/VirtualSimPresence-pdfrev.pdf

[16] G. Wong and V. Wong, "Virtual reality in space exploration." [Online]. Available: http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/kcgw/report.html

[17] R. U. Sixense Entertainment, "Razer hydra." [Online]. Available: http://www.razerzone.com/gaming-controllers/razer-hydra/

[18] S. Patel, "Durovis dive." [Online]. Available: http://www.durovis.com/opendive-howto.htm

[19] U. Technologies, "Unity3D." [Online]. Available: http://unity3d.com/

[20] S. Meyer and A. Rakotonirainy, "A survey of research on context-aware homes," p. 1–10.

[21] A. Fleury, J. S. Pedersen, and L. Bo Larsen, "Evaluating user preferences for video transfer methods from a mobile device to a TV screen," vol. 9, no. 2, pp. 228–241. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1574119212000636

[22] B. Avery, B. H. Thomas, J. Velikovsky, and W. Piekarski, "Outdoor augmented reality gaming on five dollars a day," in *Proceedings of the Sixth Australasian conference on User interface-Volume 40.* Australian Computer Society, Inc., p. 79–88. [Online]. Available: http://dl.acm.org/citation.cfm?id=1082254

[23] M. J. Schuemie, P. Van Der Straaten, M. Krijn, and C. A. Van Der Mast, "Research on presence in virtual reality: A survey," vol. 4, no. 2, p. 183–201. [Online]. Available: http://online.liebertpub.com/doi/abs/10.1089/109493101300117884

[24] J. L. Silva, J. Campos, and M. Harrison, "Formal analysis of ubiquitous computing environments through the APEX framework," in *Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems*

- *EICS '12*. New York, New York, USA: ACM Press, 2012, p. 131. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2305484.2305506

[25] A. Oulasvirta, E. Kurvinen, and T. Kankainen, "Understanding contexts by being there: case studies in bodystorming," pp. 125–134, 2003.

[26] D. Wigdor and D. Wixon, *Brave NUI World*.

[27] "Zotero | home." [Online]. Available: http://www.zotero.org/

[28] D. A. Norman, *The Design of Everyday Things: Revised and Expanded Edition*. Basic Books. [Online]. Available: http://books.google.se/books?id=nVQPAAAAQBAJ

[29] I. Oculus VR, "Oculus rift." [Online]. Available: http://www.oculusvr.com/

[30] I. Leap Motion, "Leap motion." [Online]. Available: https://www.leapmotion.com

[31] SmartWatch - sony smartphones. [Online]. Available: http://www.sonymobile.com/se/products/accessories/smartwatch/

[32] 5Dt DataGlove 5 ultra. [Online]. Available: http://www.5dt.com/?page_id=34

[33] B. Laboratories and A. J. B. Brush, "Home computing unplugged : Why , where and when people use different connected devices at home," p. 627–636.

[34] "Human interface guidelines v1.8.0." [Online]. Available: http://go.microsoft.com/fwlink/?LinkID=247735

[35] Y. Ishiguro, A. Mujibiya, T. Miyaki, and J. Rekimoto, "Aided eyes : Eye activity sensing for daily life." [Online]. Available: http://www.vision.cs.chubu.ac.jp/CV-R/pdf/RekimotoACM2010.pdf

[36] S. Pescarin, E. Pietroni, L. Rescic, K. Omar, M. Wallergård, and C. Rufa, "Nich: A preliminary theoretical study on natural interaction applied to cultural heritage contexts," in *Digital Heritage International Congress (DigitalHeritage), 2013*, vol. 1, Oct 2013, pp. 355–362.

[37] I. Oculus VR, "Oculus SDK." [Online]. Available: https://developer.oculusvr.com/

[38] "Graphics cards - GTX680-DC2T-2GD5." [Online]. Available: http://www.asus.com/Graphics_Cards/GTX680DC2T2GD5/overview/

[39] Android debug brigde. [Online]. Available: http://developer.android.com/tools/help/adb.html

[40] I. Autodesk, "Autodesk 3dsMax." [Online]. Available: http://www.autodesk.se/products/autodesk-3ds-max/overview

[41] "3d models, textures and plugins at turbosquid." [Online]. Available: http://www.turbosquid.com

[42] Logcat. [Online]. Available: http://developer.android.com/tools/help/logcat.html

[43] "GIMP imaging editor." [Online]. Available: http://www.gimp.org/

[44] "FFmpeg." [Online]. Available: http://www.ffmpeg.org/

[45] D. Kirsh, *Inverse Kinematics*. [Online]. Available: https://www.assetstore.unity3d.com/#/content/1829

[46] LeanTween. [Online]. Available: https://www.assetstore.unity3d.com/#/content/3595

[47] I. Autodesk, "Project pinocchio." [Online]. Available: http://projectpinocchio.autodesk.com/

[48] S. Studios, "Sixense unity plugin." [Online]. Available: https://www.assetstore.unity3d.com/#/content/7953

[49] NASA, "Nasa tlx paper/pencil version." [Online]. Available: http://humansystems.arc.nasa.gov/groups/tlx/paperpencil.html

[50] S. Carter and J. Mankoff, "Prototypes in the wild lessons from three ubicomp systems," *Pervasive Computing, IEEE*, pp. 51–57, 2005. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1541968

[51] T. Grill, O. Polacek, and M. Tscheligi, "Conwiz: A tool supporting contextual wizard of oz simulation," in *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*, ser. MUM '12. New York, NY, USA: ACM, 2012, pp. 21:1–21:8. [Online]. Available: http://doi.acm.org/10.1145/2406367.2406394

[52] D. Courter, J. P. Springer, C. Neumann, C. Cruz-Neira, and D. Reiners, "Beyond desktop point and click: Immersive walkthrough of aerospace structures," in *2010 IEEE Aerospace Conference*. IEEE, Mar. 2010, pp. 1–8. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5446793

[53] C. Cruz-Neira, D. J. Sandin, T. A. DeFanti, R. V. Kenyon, and J. C. Hart, "The cave: Audio visual experience automatic virtual environment," *Commun. ACM*, vol. 35, no. 6, pp. 64–72, Jun. 1992. [Online]. Available: http://doi.acm.org/10.1145/129888.129892

[54] K. Hornik, "R FAQ," 2014. [Online]. Available: http://CRAN.R-project.org/doc/FAQ/R-FAQ.html

[55] H. Wickham, "ggplot2," 2013. [Online]. Available: http://ggplot2.org/

[56] D. A. Norman, "Natural user interfaces are not natural," vol. 17, no. 3, p. 6–10. [Online]. Available: http://dl.acm.org/citation.cfm?id=1744163

[57] "Announcing the Oculus Rift Development Kit 2 (DK2)." [Online]. Available: http://www.oculusvr.com/blog/announcing-the-oculus-rift-development-kit-2-dk2/

Appendix A

Appendix

A.1 Concepts description

This appendices contains a table that represents how the concepts developed during the design phase were divided into four categories and three time periods. The categories are:

- *Device Discovery* – How are devices and services discovered? In what time frame is the concept plausible?
- *Device Selection* – How are devices and services selected? In what time frame is the concept plausible?
- *Select UI Output* – Where are users presented with the UI of the device/ service? In what time frame is the concept plausible?
- *Controlled UI* – Where do users interact with the UI of the device/ service? In what time frame is the concept plausible?

Table A.1: All the concepts that passed the first quality gate

	Device Discovery	**Device Selection**	**Select UI Output**	**Controller UI**
Now > 2014	-Categorical Perception - UPnP Services	-Button Input	-On Device UI -App Based UI	-Physical remote controllers/ terminals -App Based Controllers
Soon 2015-2019	-Natural gestures -Image Analysis -Notifications -Augmented descriptions	-Touch to interact -Handshake to grant permissions -Selection by saying intent -Virtual presence	-One controller for all devices	-Gesture control -Behind the glass interaction -3D sensing
Future > 2019	-Natural language to find device -Area Clouds -Indoor Positioning	-Grab gesture to select -Selection by pushing content -Card-based interaction -Eye tracking -Context based	-Augmented UI -No UI -Smart textiles	-Advanced speech control -Tangible interfaces with haptics

Rejected Concepts

Two Hi-Fi prototypes that were rejected are presented below. Some other notable concepts that were quickly implemented and rejected were:

- A Mid/Hi-Fi prototype for -Image Analysis was built in the Unity3D Game Engine with a Qualcom Vuforia plugin. How this this prototype looked can be seen in Figure A.1. See-through Augmented Reality with computer vision seemed like a good way to showcase this concept. However it was rejected since a decision was made to only focus on the demoing environment and concepts created within it. This in order to strive for better, more full fledged concept implementations and to easier make comparison between the different concepts.

- A Mid/Hi-Fi prototype for -Selection by saying intent was developed for Android and the Sony SmartWatch. The implementation utilized `android.speech.SpeechRecognizer` and `Sony Add-on SDK`. This was done to showcase how wearables might be used in the future. The benefit of using a watch or any similar technology over any other input device lies in its form factor. For example, should a wrist attached wearable device be coupled with location and direction capabilities, the system could gain context about what device the user points towards. The purpose of rejection was the same as the previous rejected -Image Analysis concept. Since there isn't any microphone on the Sony SmartWatch an Android phone was used to input speech. How this prototype looked can be seen in Figure A.2.

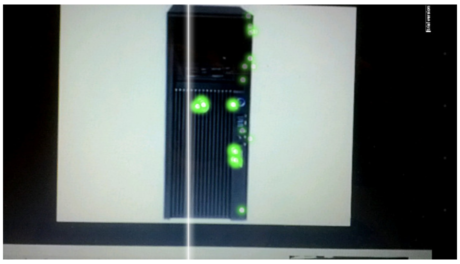

Figure A.1: -Image Analysis

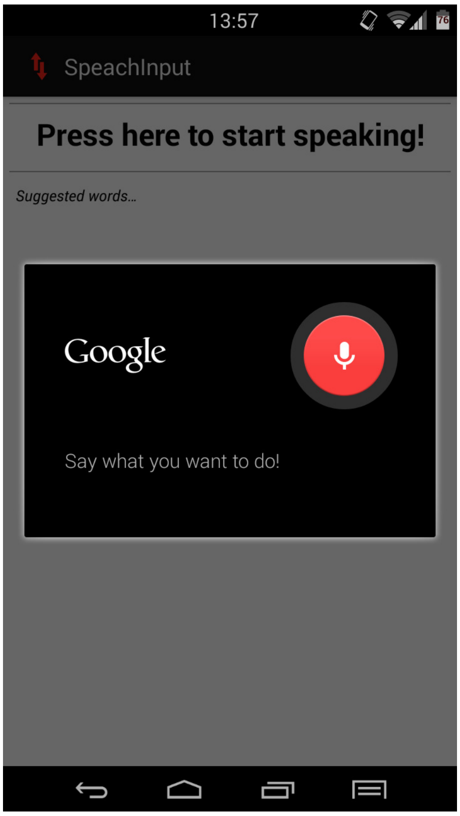

Figure A.2: -Selection by saying intent

Device discovery

Now

Categorical perception: Discovering devices with categorical perception basically means analysing your surroundings – starting complex cognitive processes that map and categorise your surroundings, supporting your ability to identify objects. The fact that most humans know the distinction between a TV and a digital photo frame is largely affiliated to their categorical perception, meaning they learn over time to distinguish between above mentioned devices. The brain is generally the best computing device out there but too much to process can increase the overall cognitive load.

UPnP services: Device discovery can be simplified by connecting a bunch of devices to an UPnP service. UPnP services allow different devices to connect to a network, also allowing users to view what devices are connected. This can reduce the cognitive load for a user that has no prior knowledge about what devices are present in a specific room/area. A big disadvantage is the fact there is no natural mapping between the devices displayed and their placement.

Soon

Natural gesture to find device: A user that has no prior knowledge of the devices in a room/area should be able to perform a "searching gesture". This gesture resembles to the kind of swiping gesture you would do on your desk when you are looking for something. Example: A user performs an in-air swiping gesture, looking for devices to interact with. When he puts his hand over a device, he is given tactile feedback (a vibration) through his smart watch, smart phone etc.

Image analysis: Wearable computing is on the up rise. If computing capabilities and a camera are given in something you wear, a new way to support device discovery can be introduced through image analysis. Example: A user wearing visors walks into a room that he has no prior connection to. His visors will start analysing his surroundings to detect if there are any consumer electronic devices to interact with. The system will in some way notify the user after analysing the picture.

Notifications: Many of the most common mobile devices have the ability to notify their user on specific events. For instance, when receiving an email the smart phone will notify you in a menu or dialogue box, pop up etc. If our mobile devices are communicating with consumer electronics in our surroundings, similar notification options can be given to support the user when discovering devices. Example: You walk into your kitchen. The coffee machine notifies you through the mobile device you're wearing, inviting you to make a cup of coffee.

Augmented descriptions: Putting an augmented tag on a device and with this tag telling the user what kind of device it is and what actions can be performed, can significantly simplify device discovery. Textual and visual tags guide the user when he chooses a device to interact with. This is especially useful when all devices are not visible. Example: A gaming console in a room is hidden inside the drawer of a TV bench. Augmenting a tag straight above it offers a user guidance if his intent was to interact with the gaming console.

Future

Natural language to find device: Natural language is very precise and a natural extension of the human mind, thus when the technology is ready for natural language input a system can pick up the users query about what devices are available in the current room and giving feedback to the user about them. Example: A user walks into a room he/ she never has been in before, thus the user has none whatsoever way of knowing what different devices are available in that room. A simple query from the user -"What devices are available here?" or -"What are my possible ways of playing the content from my cellphone in this room?" could be a non intrusive and powerful way for the user to gain knowledge about what his/ hers possibilities are.

Area clouds: Imagine in the future that all our consumer electronics are interconnected and as well connected to a cloud service where all users content and information about the users currently owned CE's are stored. This would enable a much more device independent and more services/ content driven paradigm. Users could hold their "virtual presence" inside a device that is personal such as a cellphone or a smart watch. The "virtual presence" holds information about positioning and can connect to connected devices in the near vicinity, thus area clouds. Example: Area clouds would enable the content to be context driven as well as aforementioned benefits. A use case could be; when a user is positioned in a sports bar where there is five connected televisions where the user is currently viewing a sports event. Should the user feel the need to walk from the sports bar the content that is of importance to the user could follow him/ her to the next place he walks, so that when the user gets on the bus from the city the sports event would be shown on his/ hers smart phone or visors and when the user gets home and walks into the living room the sports event will instantaneously be shown on the wall mounted display.

Indoor positioning: A system that keeps track of a users position within a room/ area can help to support device discovery. If the user has an intent and wants to act in some way, the system can give him feedback on where he can perform this action - by guiding him to the appropriate device. Example: Walking into a hospital, the user feels the urge for some coffee. His position within the building is registered and depending on where he is located, the system provides information on nearby coffee machines, through any of the users mobile devices.

Device selection

Now

Button input: Selecting a device through button input is the most common way of device selection. Physically touching an on/off button as a way to select a device is in not a way bad, since it offers pretty good mapping and affordance. However it creates some unwanted difficulties and frustration. When many buttons are given it is sometimes hard to know what button to press - also since placement of this button isn't the same on all devices, the user sometimes spends a fair amount of time looking for it. Virtual buttons that represent physical devices, i.e printers, are generally bad and don't offer the same mapping and affordance as physical buttons.

Soon

Touch to interact: Selecting a device by simply touching the device that is wanted could be a powerful means of selection. The problem with todays device selection is that the natural mapping between what represents a device and what the device is in reality is in many cases non-existent. When a user sits in a office environment and wants to print a couple pages there is no natural mapping between the list and what printers the list represents. A table with a bunch of controllers (the traditional ones) offers no mapping whatsoever. Example: Imagine that the user simply prints and then walks up to a preferred printer upon touching a preferred printer that printer is chosen as the current printer for the job and the printing begins.

Handshake to grant permissions: To grant a friend access to for example a wireless network is today often a problem. Sometimes in private networks connected devices share services that users do not want to grant others access to. This is sometimes solved by a private and a public part of the network, this also has limits such as the password part et al. By granting access/ permissions with a handshake would be a natural way of solving the password part, in smart systems this could be combined with one or more selected devices. The handshake is in western culture a means of greeting acquaintances and in extension it is a means of saying "I trust you", culturally it has been used in business transactions as a de facto standard of agreeing. Using this to grant permissions would be a natural extension. Example: Imagine that the users friend comes over for a coffee break, they greet by shaking hand and temporal permissions for "safe" interactions are granted. Now the friend has the means to play his/ hers content on the television, stereo set et al.

Selection by saying intent: When we talk, we can be as precise as we want. Talking is generally the best way of telling your surroundings what you aim to do. Having a system which listens to and recognises speech can be made into a natural way for users to select the device they would like to interact with. Example: A user with a specific intent walks into a room with a bunch of devices.

Traditionally he would use a controller/terminal, entering button input to start interacting with the device/s that suit his intent. Instead of doing this he now says what he wants to do and the device/s are made available to him, directly starting the service corresponding to the users intent.

Virtual presence: As aforementioned in Area clouds, virtual presence is the user's current position coupled with the devices that he/ she is currently connected to as well as his/ hers identity. The virtual presence could be held in any personal device that seldom leaves the person, such as a cellphone or a smart watch.

Grab-gesture to select: When people want to interact with physical entities they pick them up, think if this would be virtually mappable onto devices. When indoor positioning is mature enough it's easy to think that it would be possible to select consumer electronics by extending an arm towards the intended CE and by grabbing in thin air towards that CE that CE would be selected as the one to interact with. Coupling this concept with a smart watch, visual and tactile feedback can be given the user both about if a device is in the direction the hand is extended and as well if the grabbed device was the intended one. Example: A user wants to watch tv, he extends an arm towards the television. Suddenly a light buzz is felt on the arm. The user closes his/ her hand still in the same direction. The smartwatch displays the device name. Should it be the correct device further interaction can be done. Should it not be the correct device the user simply does a throwing away gesture and thus throws away the device.

Selection by pushing content/intent to device: All consumer electronic devices are affiliated to a set of actions that can be performed on them. For instance; a printer is used for printing documents, a digital photo frame is used for displaying pictures, you play games on your gaming console and watch movies on a TV. Device selection can be simplified by simply choosing content and pushing it onto a device whose actions correspond to the users intent. It resembles to the everyday putting-content-into-device-notion (i.e putting a blu ray disc into a blu ray reader). Example: A user is sitting in his office and writing a document on his tablet. When he's done he pushes the document towards his printer. The printer receives the content (document) and starts its main activity, printing.

Card-based interaction: As mentioned in One controller for all devices there is both a need for great device selection and great ways of coupling UI with device. This could be done in a manner similar with a business card analogy. Usually when a person divulges the possible ways of reaching him/ her to another there is often a handing out of a business card involved. This analogy could be translated into the virtual/ device realm. The devices could be divulging it's business card to possible interactors, for them to pick up and use to interact with. The cards could be virtual or real and they should contain the means of how to interact with their corresponding CE and what content they can take part of.

This would enable users to “push” content to devices where they have access to a business card.

Selection with eye tracking: An interaction concept that has been thoroughly researched and which many sees as the next step in interaction is eye tracking and eye gazing. To couple eye tracking/ gazing with other interaction methods could make for great device selection. Using eye tracking/ gazing for device selection would be a natural extension of everyday interaction with people. Example: A user sits in a office environment. Suddenly he/ she feels the urge to show the neighbour a snippet. Instead of calling the neighbour over, the user marks the snippet looks at the neighbours screen and sends over the snippet.

Context based selection: Combining a user’s context and location can act as a tool for smart systems to determine what device to make available to the user. Smart systems that over time learn what a users does at a given time and place, will enable seamless device selection. Example: Imagine you are booked to travel from Lund to Kristianstad. Since you’ve put this event into your calendar, the system will know what you are about to do. When arriving at the train station the system will notify the ticketing machine to prompt you to buy a ticket to Kristianstad. When you’ve picked up your ticket and started walking towards the platform area, a public display near you will display the departure time of your train, the correct platform and possible delays.

Select UI output

Now

On-device UI: The paradigm of today has more or less been to have the UI on the device that is interacted with at the moment. This has a lot of disadvantages such as when a television set is connected to a home movie system. When the volume is to be raised, do you raise the volume on the tv or on the receiver? Since receivers often have a much smaller display and oftenly are situated so that they are not seen from the majority of the room it is hard to have visual feedback about the level of the volume.

App-based UI: For example, in Windows 7 when you wish to alter the volume of anything the user is greeted with this volume mixer that adds all services that outputs sound. Sometimes this is the view that the users want but it could be very burdensome and sometimes the users doesn’t remember what settings they set in what program. It also often leads to unnecessary troubleshooting.

Soon

One controller for all devices The problem of today is the chaos that is created by all the remote controllers in peoples living rooms. On a normal living room table in Sweden it is not unnatural to have more than five controllers, as people gets more and more consumer electronics the amount of remote controllers will explode. There is seldom good mapping between controller and device which leads to high mental workload. Instead it would be much better if the users could use for example their cellphone to interact with all their consumer electronics. For this to work well there needs to be a good way of doing device selection so that you as a user knows what device you're about to control.

Future

Augmented UI (controllers everywhere): Today user interfaces are coupled with a specific device, meaning they probably are shown on that device. By augmenting the user interface a more flexible system is built, enabling a user situated in the dining area of his kitchen to control the dishwasher situated in a different area. Being able to control a device anywhere makes for a smart environment where users intents are not limited to a room/area. Example: Doing your laundry and watching a movie are two different activities that both require a users attention. When watching a movie you usually don't want to be interrupted so doing your laundry at the same time can be a bit tricky (since you might have to leave the room). Augmenting the washing machines UI, a user can entirely focus on the movie but at the same time have an overview over his laundry.

No UI: Why do users need UI's, as of today we have all been acting on UI's shouldn't we instead be interacting with the content. Having the content as the UI would enable full transparency of interaction and action.

Controllers in textiles: We are surrounded by textiles: the clothes we wear, curtains, our couch etc. Rendering the UI output in textiles could be a good way of making interaction with CEs available anywhere. Also, this kind of interaction would minimise the steps between a user's intent and the performing of this intent. Example: A user wants to quickly alter the settings for a CE. He selects this CE and outputs the UI on his couch, doing what he wants to do in a quick manner.

Controlled UI

Now

Physical remote controller/ terminals: As of today much of the interaction we as users perform with our devices is performed on a third entity. We are not performing the action on the content, and we are neither acting on ourselves. Instead we perform action on remote controllers and or terminals. This often leads to poor mapping between intent and action and yields high mental workloads for the user.

App-based controllers: App representations of physical controllers are getting more common. They use the mobility of the smart phone/ tablet to "get rid of" physical controllers. However they do acquire additional software, the app itself, to run. Another problem is the fact that an app sometimes can be hidden within the system its installed on - meaning users would need to find the controller before actually controlling the CEs they represent. Thus we have almost the same situation as with the physical remote controller.

Soon

Gesture control: User interfaces using gesture input is already fairly common. Some video players utilise kinect like sensors for functions such as playing/ pausing and rewinding. Also, some smart phones let you do in-air gestures, above the device, enabling scrolling and similar functionality. A big problem with UIs that can be controlled with gestures is the fact that there are many possibilities for false positives. Scratching your head can in other words have unwanted consequences if a similar gesture is implemented in the UI for a device you are interacting with. To come to its right, gesture controlled UIs need to scale down the amount of gestures used. Also, using our mobile devices we can limit gesture control: if the user is wearing a smartwatch he can only perform a gesture with that hand. These limitations can help to solve the problem with false positives. Example: Given a UI, either augmented, behind the glass or in some other way, the user uses his arm with a smart watch strapped to it to perform a vertical movement, scrolling within the user interface.

Behind the glass interaction (touch input): The advantages with behind the glass interaction has been made widely known to the large masses since the smart phone revolution. Behind the glass interaction is considered to be very intuitive and is, as of today, one of the most common ways of interacting with user interfaces. Example: Using either our most common mobile devices, the smart phone, tablet and smart watch, or any other device with glass (why not simply a piece of glass or a glass table) the user interface for a consumer electronic device can be presented within these controller devices. This enables the user to interact with his CE of choice in a intuitive and well known matter.

3D sensing Similar to gesture control, 3D sensing uses the users gestures to control the UI.

The main difference is that with 3D sensing the user doesn't need to perform the gestures near, either a controlling device or the device being controller, but in a free matter in the room.

Future

Advanced speech control: The main advantage with speech is its preciseness. User interfaces with speech control that let the user have a dialogue with the system can be a good way of controlling user interfaces. The main disadvantage is the fact that speaking to consumer electronics is not really socially accepted. And there needs to be good affordance for what possible interactions are possible. This can be helped if the speech control is smart enough, enabling good dialog and aiding the user in their interactions. Example: The user is sitting in his workplace, writing a bunch of documents. A printer is situated a few meters away. When he's done writing he tells the printer to print the document he wrote. Since he wrote a bunch of documents the printer can ask if the user wants to print document A, B, C The user tells the printer which of the documents to print and the printer does just this.

Tangible interface with haptic feedback: If the user interface of a CE is mapped to a physical thing, a more hands-on way of interacting with UIs can be introduced. Changing shape and using vibrations for instance, the physical thing can offer the user haptic feedback on the action he is performing. Example: Using a ruler-like thing with a slider, a user can alter the volume setting on his T V. Sliding the slider to the right would imply the volume up function. If the thing also offers haptic feedback it could vibrate, increasing the vibration when the volume is turned up.

A.2 Usability test - Script

With UbiComp we are exploring technology and interaction beyond the desktop environment. The aim is to seamlessly connect users, mobile devices and consumer electronics. To be able to do this the environment is given perceptional capabilities. It is aware of the users context and can interpret what his intention is by monitoring the implicit actions a user does. The vision is that users should be able to interact with their consumer electronics in a more intuitive and direct manner: by talking, moving and gesturing. This means that users can concentrate on their task rather than being forced to explicitly interact with a computerized system.

The aim of our work is to bridge over from the current interaction paradigm, where users are forced to explicitly interact with computerized systems, to more ubiquitous interactions.

Background Our thesis is entitled ”Ubiquitous computing in a home environment - Controlling consumer electronics”. The idea is to develop a number of interaction concepts and evaluate them. That's why YOU are here. With our interaction concept we say that we want to simplify the discovery of and the interaction with consumer electronics, with the help of our mobile devices.

How is this done? Well, it is here ubiquitous computing comes into the picture. The idea behind ubiquitous computing is to explore technology and, above all, interaction beyond the desktop environment. The goal is to seamlessly connect users and technology - linking the physical world with the digital.

This is done by creating an environment with built-in intelligence: that is aware of the user's context and has the ability to interpret the user's implicit actions. What then is an implicit action? Well, an implicit action is an action performed by the user that is not primarily aimed to interact with a computerized system but which such a system understands as input.

An example to clarify this can be when you drive your car to your garage. Imagine that the garage door is controlled by a computerized system with built-in intelligence, that is aware of users' context and can interpret implicit actions. Your context is that you're on your way home from work and your implicit action is that you drive up towards the garage door. If the computer system is aware of your context and can interpret your implicit actions as input, then the garage door can automatically open without you having to interact with the computer system that controls garage door.

How do our mobile devices come into the picture? We told you earlier that our goal was to simplify discovery of and interaction with consumer electronics, with the help of our mobile devices. What we say in our work is that the mobile devices we use can help us to connect the physical world with the digital, by acting as sensors in the physical environment. The mobile devices can build our context, keep track of our implicit actions and convey all of this to the intelligent environment the user is situated in.

The vision is that users should be able to interact with a system or environment in a more intuitive way: by talking, moving and gesturing. User should, in other words, not be distracted by being forced to explicitly interact with the system. It is often said that ubiquitous computing is the new interaction paradigm. The aim of our work is to bridge the current interaction paradigm, where users constantly are forced to explicitly interact with different systems to a more ubiquitous interactions.

Our concept we have created in a virtual environment , with built-in intelligence , which you as a test person sitting in a living room and perform a simple scenario. The scenario we will describe in a moment.

Explaining the setup Now you will do a test run of the virtual environment to get familiar with it. We use various input methods for mapping your movements in the physical environment to a character in the virtual environment.

The Razer Hydra is used for arm tracking and the 5DT Gloves are for finger tracking.

You will be wearing the Oculus Rift and when you wear it, you will get a stereoscopic image of the environment. It will feel as if you are the virtual character. The Oculus Rift is also used to simulate a pair of visors. These make it possible to augment or project virtual objects in the physical environment.

You will be interacting with the tablet in front of you in two tests. What you do on it is linked to events in the virtual environment. We say that the tablet is *yours* and it carries your digital identity. It tells the environment what services you use and what videos, music and other content you own.

The smartphone will be strapped to your wrist and it is used to simulate a smart bracelet or a smart watch. The smart bracelet / watch tells the system who you are, what your context is. It also keeps track of your implicit actions.

A.3 Usability test - Questions

An unstructured interview was conducted after each test. A number of questions were asked to spark the discussion. The *general questions* were aked after each test, while the *test specific questions* were asked for a specific test. After the test session, three questions were asked as a debriefing.

General questions

1. What are your initial thoughts on this interaction concept?
2. What went well?
3. What went worse?
4. Do you have any general thoughts you would like to share?

Test specific questions

- Gaze
    1. Are there any specific use cases where using ones eyes to the detect devices could be beneficial to a user?
- Gesture
    1. Are there any specific use cases where using a gesture to the detect devices could be beneficial to a user?
- Push *(question asked after both Push and Grab had been tested)*
    1. Push is content-centered, while Grab is device-centered. What is more logical, to first decide what content you would like to interact with or what device that content should be displayed on?
- Grab *(question asked after both Grab and Push had been tested)*
    1. Grab is device-centered, while Push is content-centered. What is more logical, to first decide what device content should be displayed on or what content you would like to interact with?

Questions after the test

1. Are there any difficulties when enabling gesture interaction?
2. How comfortable are you sharing your context and personal information to a 'smart environment'?
3. How would you like to interact with consumer electronics in the future?

A.4 TLX Weight

In order to asses the interaction concepts a NASA TLX was used. It is a tool used for rating perceived workload and consists of six sub-scales.

The six sub-scales are compared pairwise, based on their perceived importance, in order to create an individual weighting. The number of time each is chosen is the weighted score. In Table A.2 the weighted score for each sub-scale is presented. The pairwise comparisons are done by the test moderator prior to the test and the same values are used for every individual calculation.

Table A.2: NASA TLX Weighting values

	Mental Demand	Physical Demand	Temporal Demand	Performance	Effort	Frustration
Gaze	3	4	1	0	4	3
Gesture	3	4	1	0	4	3
Push	4	2	1	0	4	4
Grab	4	2	1	0	4	4