

2015-01-08

Remote Control Design for a Ubiquitous Computing Ecology

Ledo, David

Ledo, D. (2015). Remote Control Design for a Ubiquitous Computing Ecology (Master's thesis, University of Calgary, Calgary, Canada). Retrieved from <https://prism.ucalgary.ca>. doi:10.11575/PRISM/27623
<http://hdl.handle.net/11023/1995>

Downloaded from PRISM Repository, University of Calgary

THE UNIVERSITY OF CALGARY

Remote Control Design for
a Ubiquitous Computing Ecology

by

David Ledo

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

CALGARY, ALBERTA

DECEMBER, 2014

© DAVID LEDO 2014

Abstract

Appliances can facilitate people’s interaction with them by outsourcing their inputs and outputs to remote controls. Remote controls can compensate for constraints in an appliance’s form factor, lessen overall cost, and enable distance interactions. Modern “smart appliances”, which can interconnect with other computational devices, take this one step further: a mobile device can control multiple appliances via custom interfaces with rich interaction capabilities. We foresee ubiquitous computing ecologies, where a room may have myriads of smart appliances all potentially controllable via a mobile device. However, this leads to four problems. It is difficult to: (1) *discover* which appliances are controllable; (2) *select* an individual appliance from the ecology; (3) *view* information about an appliance; and (4) pertinently reveal *controls*. We mitigate these problems by applying the theoretical concepts of *proxemic interaction* and *gradual engagement* to the design of mobile remote controls. In particular, our remote control designs mimic social protocols in which people orient towards and approach one another to mediate interpersonal interactions, except that in our case we mediate person to appliance interaction. This thesis covers and contributes a design exploration and prototype that demonstrates our application of these concepts.

Publications

Some materials, ideas and figures from this thesis have appeared previously in the following publications:

David Ledo, Saul Greenberg: Mobile proxemic awareness and control: exploring the design space for interaction with a single appliance (2013). In *CHI '13 Extended Abstracts on Human Factors in Computing Systems – ACM CHI 2013 Video Program* (Paris, France). ACM April 27 - May 2, Pages: 2831-2832.

Acknowledgements

This thesis would not be what it is had it not been for the help and support from many people. My M.Sc. may come across as just a milestone, but I will always treasure it as a formative experience that truly initiated me into HCI research.

First of all, I would like to thank my supervisor, Saul Greenberg for getting me to this point. Saul: I learned a lot in these past two years, and I think that this thesis is just a small instance of this journey we went through together. Thanks for always being there, watching me grow every step of the process and for all your help: from being my course instructor and teaching me about the value of *small but interesting problems*, to the numerous discussions about sketches, checking out my demos and coming up with a new option to look at, teaching me about the role of theory in designing interactive systems all the way to how to write chapter 1. I think you gave me a very complete experience that has given me what I consider a good sense of how to do research. I would also like to thank my examiners Ehud Sharlin, Frank Maurer and Barry Wylant. Their comments and suggestions made this thesis into a much stronger piece of writing.

Next I would like to thank those who will be my official co-authors when this thesis gets written up for a conference: Nicolai Marquardt and Sebastian Boring. You guys are an inspiration and I look forward to writing with you again. Thanks for all the discussions (in person and online) and the tech support for moments where I was stuck.

The results of my Masters thesis wouldn't be what it is without the love and support of my girlfriend Lindsay MacDonald. Lindsay, thanks for being there from day one and being my pillar of support. Thanks for helping me with little things like finding ways to keep marker sticks together and killing my hesitation whenever I wasn't sure about a figure, all the way to proof-reading my chapters and telling me how to make my work better. You make me a better person.

My friends and colleagues were also of great help. I would like to thank Lora Oehlberg and Jagoda Walny for proof-reading my chapters and the super interesting discussions that we had. Lora: thanks for pointing me towards design methods that could help me frame what I'd done; I really look forward to working more with you. Jagoda: thanks for helping me think deeper about my project and find out better ways to motivate my work. My peers who embarked on the M.Sc. at the same time as me: Setareh Aghel-Manesh, Jiannan Li, Richard Tang, Hesam Alizadeh and Bon Adriel Aseniero – it was great to know that I wasn't alone in this journey, and I hope that I have been as supportive as you guys have been. Frederik Brudy, thanks for helping me setting up the home space into the awesome space it became, and for the dynamism getting things done. The other lab professors – Sheelagh Carpendale and Tony Tang – you guys taught me a lot, and I appreciate all the 'research-lite' discussions that we had. Finally, I would like to thank all the current and past members of the iLab that have come and gone over time. It's incredible how supportive a workplace can be and how it can pull the best out of people.

Last but not least, I want to thank my family: my mom (Maria Angelica), my dad (David) and my sister (Michelle). You guys have been there from the beginning and have made me who I am today. You've heard me out constantly and even helped me build the first lamp prototype. I hope this thesis serves as proof of our mutual support and the high achiever I have become thanks to you.

Table of Contents

| | |
|---|-------------|
| Abstract | ii |
| Publications..... | iii |
| Acknowledgements | iv |
| Table of Contents | vi |
| List of Figures | x |
| Video Figures..... | xiii |
| Chapter 1. Introduction | 1 |
| 1.1 Motivation – Ubicomp Ecologies..... | 2 |
| 1.2 Problem Statement | 8 |
| 1.3 Approach and Research Objectives | 11 |
| 1.4 Methodology..... | 13 |
| 1.5 Research Context and Scope | 15 |
| 1.6 Thesis Structure | 15 |
| Chapter 2. Background and Related Work | 17 |
| 2.1 Remote Controls Yesterday and Today..... | 18 |
| 2.2 Appliances in Ubicomp Ecologies | 25 |
| 2.2.1 Ubicomp Ecologies | 25 |
| 2.2.2 Ubiquitous Appliances..... | 26 |
| 2.2.3 Discovering and Selecting Appliances in a Ubicomp Ecology | 27 |
| 2.3 Proxemic Interaction and Gradual Engagement | 37 |
| Chapter 3. Conceptual Framework | 41 |
| 3.1 An Ecology of Appliances | 41 |
| 3.1.1 Appliance Mobility..... | 43 |

| | | |
|-------------------|--|-----------|
| 3.1.2 | Directness of Appliance Interaction | 45 |
| 3.1.3 | An Appliance's Physical Manifestation | 46 |
| 3.1.4 | Appliance Groups | 46 |
| 3.1.5 | Appliance Complexity | 47 |
| 3.2 | Operationalizing Proxemic Interaction | 48 |
| 3.2.1 | Leveraging Proxemic Variables | 49 |
| 3.2.2 | Gradual Engagement of Controls | 51 |
| 3.2.3 | Presenting Information on a Mobile Device | 56 |
| Chapter 4. | Proxemic-Aware Controls in Action | 61 |
| 4.1 | The Appliance Interfaces | 61 |
| 4.1.1 | Viewing and Interacting with Appliances | 63 |
| 4.1.2 | The Thermostat | 65 |
| 4.1.3 | The Lamp | 66 |
| 4.1.4 | The Radio | 67 |
| 4.1.5 | The Router | 68 |
| 4.1.6 | The Printer | 69 |
| 4.1.7 | The RoomViewer | 70 |
| 4.1.8 | Contrasting the Appliances | 71 |
| 4.2 | Main Usage Scenarios | 71 |
| 4.2.1 | Discovering Interactive Appliances | 72 |
| 4.2.2 | Gradually Engaging with an Appliance | 75 |
| 4.2.3 | Manually Overriding Proxemics | 77 |
| 4.2.4 | Around-Appliance Menu Navigations | 79 |
| 4.2.5 | Room View Hierarchy | 80 |
| 4.2.6 | Situating Context of Actions via Proxemics | 82 |
| 4.3 | Secondary Scenarios | 83 |
| 4.3.1 | Identity-Based Access Levels | 83 |
| 4.3.2 | Extending Appliance Functionality and Leveraging Existing Controls | |

| | |
|---|------------|
| Chapter 5. System Details | 87 |
| 5.1 Prototyping Environment | 87 |
| 5.2 Visual Representation | 89 |
| 5.2.1 Changing the Interface as a Function of Proximity: Discrete versus Continuous Engagement | 89 |
| 5.2.2 Transitioning Between Controls as a Function of Orientation and Distance: Showing Objects in Space..... | 90 |
| 5.3 System Architecture | 96 |
| 5.3.1 General System Details..... | 96 |
| 5.3.2 Appliances..... | 98 |
| 5.3.3 Underlying Software Structure..... | 99 |
| 5.4 Limitations | 103 |
| 5.4.1 Scalability..... | 104 |
| 5.4.2 Configuration..... | 104 |
| 5.4.3 Rules of Engagement..... | 105 |
| 5.4.4 Architecture | 105 |
| 5.4.5 Technical..... | 107 |
| 5.4.6 Evaluation..... | 107 |
| Chapter 6. Conclusions and Future Work | 109 |
| 6.1 Revisiting Thesis Problems and Solutions..... | 109 |
| 6.2 Contributions | 112 |
| 6.2.1 Major Contributions | 113 |
| 6.2.2 Minor Contributions..... | 113 |
| 6.3 Avenues for Future Work | 113 |
| 6.3.1 Architecture that Enables Ad-Hoc Proxemic Information | 114 |
| 6.3.2 Exploring and Comparing Different Form Factors..... | 114 |
| 6.3.3 Applying Mobile Gradual Engagement to the Physical World | 115 |
| 6.3.4 Prototyping New Deployable Tracking Technologies | 116 |

| | |
|--|------------|
| 6.4 Closing Remarks..... | 116 |
| References | 118 |
| Appendix – Methodology and Process..... | 128 |
| A.1 Design Guidelines through HCI Theory | 131 |
| A.2 Concept Design | 132 |
| A.3 Generic Design..... | 133 |
| A.5 Process and Lessons Learned..... | 134 |
| A.5.1 Defining the Initial Problem Statement | 134 |
| A.5.2 First Theoretical Enquiry – Proxemic Interaction and Gradual Engagement | 134 |
| A.5.3 First Ideation Stage | 135 |
| A.4.5 Grouping and Classifying Items – First Framework | 136 |
| A.5.5 First Milestone..... | 137 |
| A.5.6 Expanding Theories and Informing the Framework..... | 138 |
| A.5.7 Augmented Reality Explorations | 139 |
| A.5.8 Towards the Final Implementation | 142 |
| A.6 Discussion | 144 |
| A.6.1 Live Demonstrations | 144 |
| A.6.2 The Role of Personal Experience | 145 |
| A.6.3 Interweaving of Design Approaches | 145 |

List of Figures

| | |
|---|----|
| Figure 1.1 Example of ubicomp ecologies..... | 2 |
| Figure 1.2 Smart appliances that are controllable from computing devices | 5 |
| Figure 1.3 Mobile interface presented by Revolv to control appliances in the home..... | 7 |
| Figure 1.4 Living room portraying the challenges of our research..... | 9 |
| Figure 1.5 Summary of this thesis' goals | 11 |
| Figure 1.6 Research context and scope. | 15 |
| Figure 2.1 Some of the first remote control technologies | 18 |
| Figure 2.2 Different remote control technologies..... | 20 |
| Figure 2.3 PDA for remote controls | 21 |
| Figure 2.4 Popular mobile remote applications | 22 |
| Figure 2.5 SmartThings..... | 24 |
| Figure 2.6 A sketch outlining ubiquitous computing ecologies | 25 |
| Figure 2.7. Scanning shows a list of connected devices | 29 |
| Figure 2.8. Scanning through video feeds | 31 |
| Figure 2.9. Spatial Referencing for discovery of devices..... | 32 |
| Figure 2.10. Physical association proposed by Want et al. | 33 |
| Figure 2.11. Work in pointing techniques..... | 35 |
| Figure 2.12. Example of proxemic distances. | 37 |
| Figure 2.13. Sequential stages presented in the gradual engagement design pattern..... | 39 |
| Figure 2.14. Example of the gradual engagement design pattern..... | 40 |
| Figure 3.1 Diagram summarizing the Proxemic-Aware Controls Framework... | 42 |
| Figure 3.2 A Spectrum categorizing appliances according to their mobility. | 43 |
| Figure 3.3 Categorizing appliances according to the directness of their interaction..... | 45 |
| Figure 3.4 Visual summary of proxemic variables | 50 |
| Figure 3.5. Gradual Engagement of Controls and its content flow. | 52 |

| | |
|--|----|
| Figure 3.6 Diagram the different types of content, which encompasses three categories: presence, state and controls – these become progressively more complex and their purpose becomes more specific. | 55 |
| Figure 3.7 Visual representation of spatial references..... | 58 |
| Figure 4.1 Appliance Rationale – this figure shows how our each of the appliances fall into different categories based on the appliance attributes described in Chapter 3..... | 62 |
| Figure 4.2 General legend for different appliance interfaces. This legend will be used in the next set of figures | 63 |
| Figure 4.3 Our system interface in action. The user holds the device and moves around the room to navigate through different controls, thus updating the interface dynamically..... | 64 |
| Figure 4.4 Thermostat interface and its progression as a function of distance. | 65 |
| Figure 4.5 Lamp interface and its progression as a function of distance..... | 66 |
| Figure 4.6 Radio interface..... | 67 |
| Figure 4.7 Router interface and its progression as a function of distance. | 68 |
| Figure 4.8 Printer interface and its progression as a function of distance. | 69 |
| Figure 4.9 RoomViewer interface showing all of the appliances in the living room. | 70 |
| Figure 4.10 Main usage scenario 1 - Discovery..... | 72 |
| Figure 4.11 Main usage scenario 2 – Gradual Engagement. | 75 |
| Figure 4.12 Main usage scenario 3 – Manual Override..... | 77 |
| Figure 4.13 Main usage scenario 4 – Around-Appliance Menus. | 79 |
| Figure 4.14 Main usage scenario 5 – RoomView Hierarchy. | 81 |
| Figure 4.15 Main usage scenario 6 – Situating Context of Actions via Proxemics. | 82 |
| Figure 4.16 Secondary usage scenario 1 – Appliance Access Levels. | 84 |
| Figure 4.17 Secondary usage scenario 2 – Extending an Appliance Functionality. | 85 |

| | |
|---|-----|
| Figure 5.1 Technical setup of our implementation. It showcases the location of the individual appliances and which computer they were connected to in order to operate. | 88 |
| Figure 5.2 First iteration of our lamp interface showing the discreet progression of stages as a function of proximity. | 89 |
| Figure 5.3 Field of view for an appliance as a function of distance: the angle threshold that establishes engagement increases as one approaches an appliance. | 91 |
| Figure 5.4 Object positioning on the screen as a function of orientation. This shows how we followed three different approaches: transparency, positioning and a hybrid that incorporated both approaches (added dotted line shows off-centered positioning). | 93 |
| Figure 5.5 Our original method for mapping orientation to screen positioning (linear mapping) compared to our correction mechanism (cubic mapping). | 95 |
| Figure 5.6 Implemented appliances | 98 |
| Figure 5.7 System architecture. | 99 |
| Figure 5.8 Screenshot of our NetworkVisualizer application. | 101 |
| Figure 5.9 Model of the Floor Lamp in our system | 103 |
| Figure A.1 Summary of the design process undertaken in this thesis. | 129 |
| Figure A.2 Design approaches as depicted by Wiberg and Stolterman | 130 |
| Figure A.3 Examples of early sketches in the design proces. | 135 |
| Figure A.4 Examples of sketches focusing on appliance interaction and proxemics. | 136 |
| Figure A.5 First framework – mobile proxemic awareness and control | 136 |
| Figure A.6 visual of the first milestone | 138 |
| Figure A.7 Storyboard showing the interaction flow using augmented reality | 140 |
| Figure A.8 Our implemented navigation techniques for target finding. | 141 |
| Figure A.9 Later sketches that informed the second milestone | 143 |

Video Figures

This thesis contains three video figures that illustrate some of the concepts described in many chapters.

The three video figures can be found at the following web locations:

Proxemic-Aware Control Scenarios (Chapter 4):

<http://pages.cpsc.ucalgary.ca/~dledomai/masters/video-figure-1.mp4>

Transitioning Between Controls (Chapter 5):

<http://pages.cpsc.ucalgary.ca/~dledomai/masters/video-figure-2.mp4>

CHI 2013 Video Submission:

<http://pages.cpsc.ucalgary.ca/~dledomai/masters/video-figure-3.mp4>

Chapter 1. Introduction

“[Controls have] too many buttons, too many nonessential features... and inconsistencies in even the simplest operations, such as numeric entry. No wonder that this set of six remotes has horrendous usability and annoys a certain user when he should be relaxing... Each of my remote controls has its own usability problems ... But the real usability disaster is caused by combining the six remotes into a single movie-playing user interface.”

- from “Remote Control Anarchy” by Jakob Nielsen¹

This thesis is concerned with the design of remote controls for a ubiquitous computing ecology (defined in the next section). This is done by enabling interaction with technology as established in our social protocols, with the aim that operating computer systems can become similar to how we naturally interact with people. As such, my thesis is that **we can leverage mobile devices and socially established protocols (proxemics) as a way to seamlessly reveal information and controls for different appliances in a room.** This chapter outlines the vision for ubiquitous computing and the relevance of appliances in this context (§1.1). I then explain the problem resulting from having an increasing number of appliances that can be remotely controlled (§1.2) and establish the goals and approach that contextualize this research (§1.3), its

¹ <http://www.nngroup.com/articles/remote-control-anarchy/>

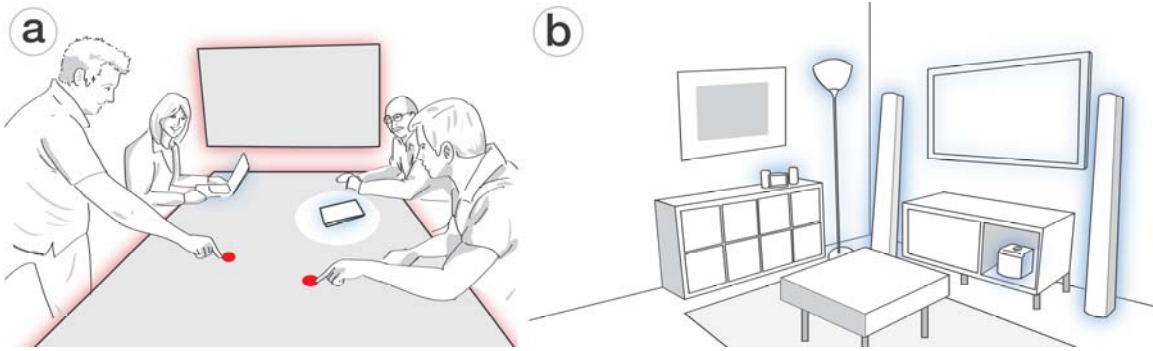


Figure 1.1 Example of ubicomp ecologies: **(a)** shows a high tech meeting room with many digital devices (laptop and digital surfaces), **(b)** shows a home space with appliances (e.g. lamp, television, radio, home theater).

methodology (§1.4), as well as its scope (§1.5). Finally, I outline the rest of this thesis and the subsequent chapters (§1.6).

1.1 Motivation – Ubicomp Ecologies

In 1991, Mark Weiser envisioned a world where computers would become prominent and commonplace, and where there would be many different types of computational devices per person. He proposed that computers would weave themselves into the fabric of our everyday lives to the point that they would become invisible, where they would enhance the world that already exists (Weiser, 1991). As part of this vision, these devices would be specific to different tasks, while still being interconnected and capable of operating together. Weiser envisioned groups of these interconnected devices working together within a particular environment, such as a small room. These environments are known as ubiquitous computing ecologies, or *ubicomp ecologies* for short. Later, Dourish (2004) elaborated that interaction within such environments should be *embodied*. He defines *embodied interaction* as situating technology and interaction in the real world context to facilitate natural social practice. Ideally, people's interaction with computers will be seamless and will fit within existing patterns of everyday interactions.

Today, over two decades later, ubicomp ecologies comprising myriads of devices are becoming a reality. Consider the ubicomp ecology of a somewhat high-tech

meeting room. As partly seen in Figure 1.1 a, its devices may comprise personal desktop computers, mobile devices such as tablets or phones, public surfaces such as digital tabletops and large displays, and public resources such as printers, scanners, routers and the like. It may also include devices controlling the room, such as its lighting. Figure 1.1 b illustrates the home setting as a ubicomp ecology, which may include personal and family computers, game consoles, home automation such as a security system, as well as specialized appliances including a home theatre and music system.

There are several large differences between present-day ubicomp ecologies and Weiser's vision. First, most device interfaces are idiosyncratic, meaning that a person must interact with each device separately through its specialized interface. This means that people must understand where each device is, where their controls are, and how to interact with it. Second, today's systems are not inter-connected, or are connectable in a very awkward manner. Yet this notion of interconnectivity in itself is crucial: it means that people can perform tasks within their device ecology in an integrated manner. Unfortunately, this only works in limited and usually restricted circumstances, and can involve considerable effort. As a simple example, we expect desktop computers to be able to print files over a network, but this often requires both (1) knowledge and technical effort in configuring the network (and security) for device to device communication, and (2) interaction effort in terms of a person selecting and configuring the appropriate printer (often from a list of cryptic names). Because device interfaces are idiosyncratic and because interconnectivity is at best awkward, today's devices are not yet woven into the fabric of everyday life, as Weiser had hoped. This problem compounds as more esoteric devices appear. Each can have quite complex interfaces, and interconnectivity becomes even more problematic. While mobile phones, tablets and desktop computers are now widespread, people still find it difficult to perform very simple tasks between them. For example, when a mix of devices are in the same location, it should be easy to move information between them. Yet this is not the case. The devices themselves may not be technically capable of talking to one

another. Even if they are, people must go through the daunting tasks of setting up ad-hoc network connections, dealing with security issues, concerning themselves with operating system and network differences (which may limit interconnection), and navigating through complex user interfaces to both configure the connection and to make information exchange possible. The overarching problem is that the interplay between devices that people want to control can be difficult.

The kinds of devices that will populate ubicomp ecologies go far beyond those that people currently recognize as computers or computer appliances (e.g., workstations, surfaces, smart phones, printers, etc.). For example, the '*internet of things*' is a vision in which all objects will eventually become digital, each with a uniquely identifiable ID and some form of communication capabilities (Kopetz, 2011). The idea is that as long as a physical object can have a small computational component, it can bridge the gap between the physical and digital world (ibid). This is already happening: we are seeing standard appliances and fixtures becoming digitally capable. Groups such as the maker community have worked on customizing or augmenting devices for specific purposes. One example is Calgary-based coffee brewing company Phil & Sebastian, which enhanced their coffee roasters with Phidgets temperature sensors to monitor the coffee making process, shown in Figure 1.2 a². Another instance is Knitic, an open source knitting machine powered by Arduino for digital fabrication of textiles. On the other hand, research in tangible interaction has explored the creation of custom devices to communicate ambient information to people. The ambientROOM made use of room fixtures to show ambient information within a workspace, as illustrated in Figure 1.2 b, such as lighting representing human movement in the work area, or a projection of water ripples to represent activity of a distant loved one (Ishii et al., 1998). Similarly, Greenberg and Kuzuoka (1999) proposed the use of physical devices as surrogates to provide awareness of remote collaborators, such as their availability for conversation. This work was extended by Hausen et al. through StaTube (2012),

² Figure reproduced from: <http://phidgets.wordpress.com/2014/05/26/how-phil-sebastian-roast-coffee-with-phidgets/>

a physical appliance that shows the status (e.g. online, away, busy) of different online contacts. As shown in Figure 1.2 c, each light in Statube represents a different contact and their current state. These examples all require a computer to control and operate the appliance, as well as set specific configurations, such as associating the tangible device to the information being presented. As such, both the maker community and the research in tangible computing show a trend of

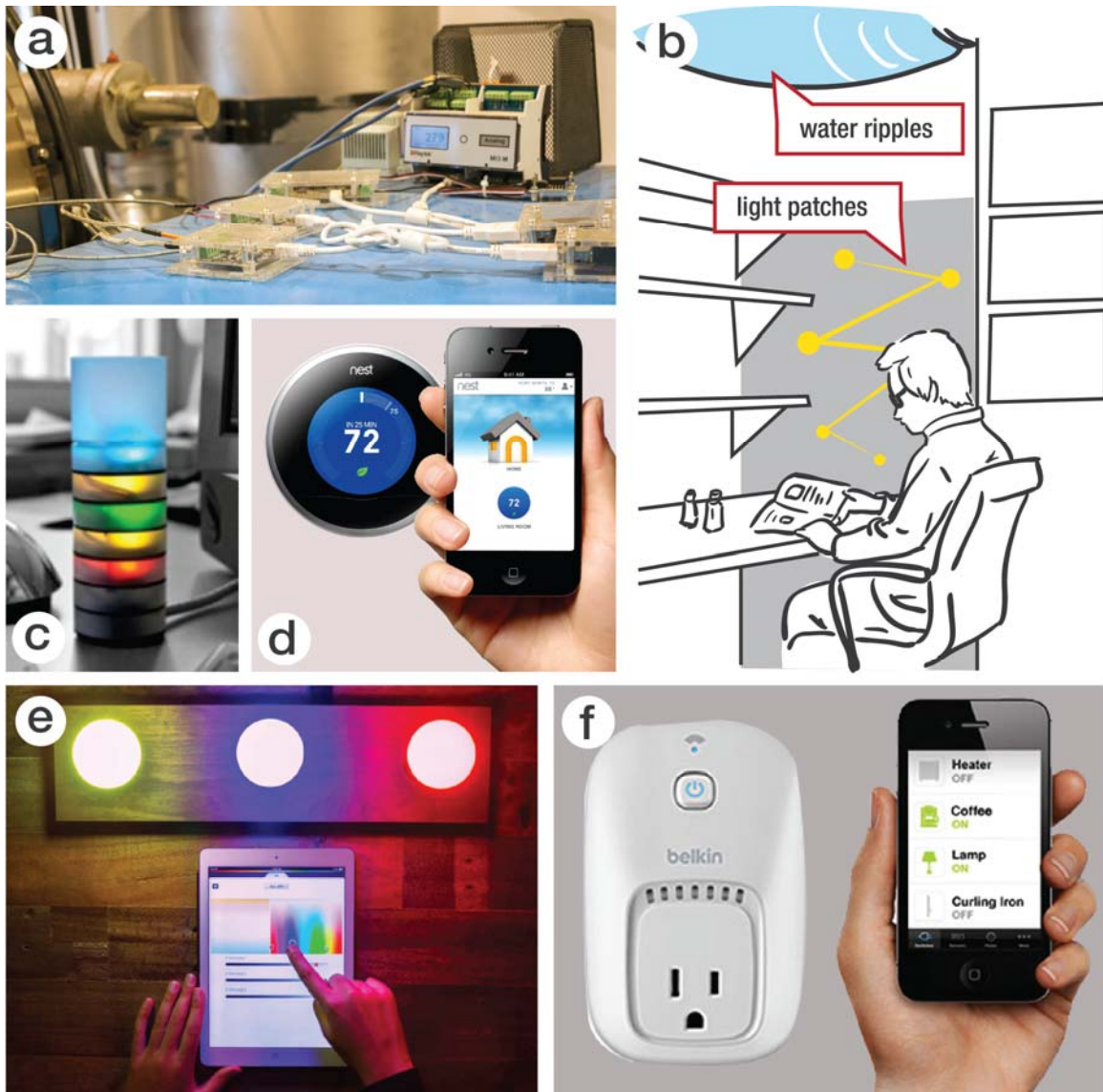


Figure 1.2 Smart appliances that are controllable from computing devices: here we see (a) a coffee roaster controlled by Phidgets; (b) the AmbientROOM; (c) Statube; (d) the Nest thermostat; (e) Philips Hue; and (f) the Belkin WeMo. *AmbientROOM sketch based on (Ishii et al., 1998), StaTube image reproduced from (Hausen, Boring, Lueling, Rodestock, & Butz, 2012)*

creating customized appliances to serve specific purposes that can be enhanced with computer-assisted controls. Nowadays, we are seeing an emergence of digitally controllable traditional appliances that are becoming commercially available. Some of these products include digital thermostats and smoke detectors (e.g. Nest³, shown in Figure 1.2 d⁴), as well as lighting systems (e.g. Phillips Hue⁵, shown in Figure 1.2 e⁶), all which can be controlled by a mobile phone through dedicated applications. Video game consoles, such as the Xbox 360, also incorporate multiple mobile devices for media control (e.g. playing movies) and information sharing through a custom network application called Xbox SmartGlass⁷.

There is no question that there will be more and more innovative devices over time. Unfortunately, this compounds the user's difficulty of controlling them. Many appliances have a small form factor, and as a result their interfaces are often quite limited or even non-existent. One solution is to use an application located on a different device to control the device of interest. This ranges from a dedicated remote controls to specialized apps located on a tablet or smart phone. Another solution offers a control center for many devices. One example is a dedicated console that controls an integrated set of known devices (e.g., a home security system). Another example are enhanced appliances (e.g., that are WiFi enabled and that obey a communication protocol) that can be hooked into control software (e.g. Belkin WeMo⁸ shown in Figure 1.2 f⁹). When different appliances follow these protocols, one can then use a commercially available control center for digitally

³ <https://nest.com/>

⁴ Figure reproduced from:
http://www.dunritehvac.net/content/images/Nest_Thermostat_iPhone.jpg

⁵ <http://www2.meethue.com/en-XX>

⁶ Figure reproduced from: <http://www.tuxboard.com/photos/2014/06/Philips-Hue-iPad.jpg>

⁷ <http://www.xbox.com/en-US/smartglass>

⁸ <http://www.belkin.com/us/p/P-F7C027/>

⁹ Figure reproduced from: <http://www.domotics.sg/wp-content/uploads/2013/04/belkin-wemo.jpg>

connected appliances (e.g. SmartThings¹⁰ and Revolv¹¹, shown in Figure 1.3¹²). These control centers all operate in a similar fashion: they make use of a specialized device capable of connecting to many of the digital devices, and they provide a direct connection to mobile devices through an app. The app presents a list of devices that it can see and enables the user to choose from this list to control that device, such as turning a light on or off. Some of the controls are rule-based, meaning that events received from one device (e.g., a motion sensor) can trigger operation of another device (e.g., having lights turn on when motion is detected).

We can see that as the number of devices in a ubicomp ecology increases, so does the difficulty of *controlling and interacting with all of these devices*. In this thesis, we explore solutions to this problem. We focus in particular on device ecologies comprising relatively simple appliances, which we broadly define as a device or piece of equipment designed to perform a specific task (vs. a multi-function general purpose computer or smart phone). These appliances of interest are commonplace. They range from small appliances (lighting fixtures, radios), major appliances (washing machine, television, heating), consumer electronics (mp3 players, digital cameras, game consoles), specialized computer components (printers, routers), and embedded automation



Figure 1.3 Mobile interface presented by Revolv to control appliances in the home.

¹⁰ <http://www.smarthings.com/>

¹¹ <http://revolv.com/>

¹² Figure reproduced from: <http://www.domotics.sg/wp-content/uploads/2013/04/belkin-wemo.jpg>

devices (thermostats, energy monitoring systems). Most of these appliances present a particular challenge for user control. Unlike desktop computers and mobile phones and tablets, they usually exhibit limited input and output mechanisms (due to spatial constraints or cost). As mentioned, controls for these appliances are usually outsourced to another device, typically a remote control, a centralized console, or an app on a mobile device. However, each incurs a cost. As the number of appliances increase, so too would the number of dedicated remote controls. This makes them impractical for ubicomp ecologies. Centralized consoles are often distant from the appliances they seek to control, and thus their use for simple operations is excessively heavyweight. Mobile devices acting as universal remote controls can potentially overcome both of these problems, but current interfaces add complexity as they require a user to manage and select between multitudes of devices. Figure 1.3 depicts this problem in a small ecology of appliances by showing a mobile interface for a universal remote (Revolv) in which each icon represents an appliance or group of appliances in the room. For instance, we can see that the ecology presented is relatively small (11 appliances), and there are three lighting fixtures that can be controlled, named “huey”, “louie” and “dewey”. Yet there is no indication which lights in the room those correspond to, whether they are on or off or what kind of actions those appliances can afford (e.g. dimming the lights or scheduling). Consequently, we can see that it is difficult to know *which* appliances can be controlled, *how* to select the appliance that we wish to control, *what* their current state is and *what* items are controllable. The next section elaborates these and other problems pertaining to the design of a remote control for a ubicomp ecology.

1.2 Problem Statement

Recall Weiser’s vision of Ubicomp as computers woven into the fabric of our everyday lives, and Dourish’s notion of embodied interaction that situates

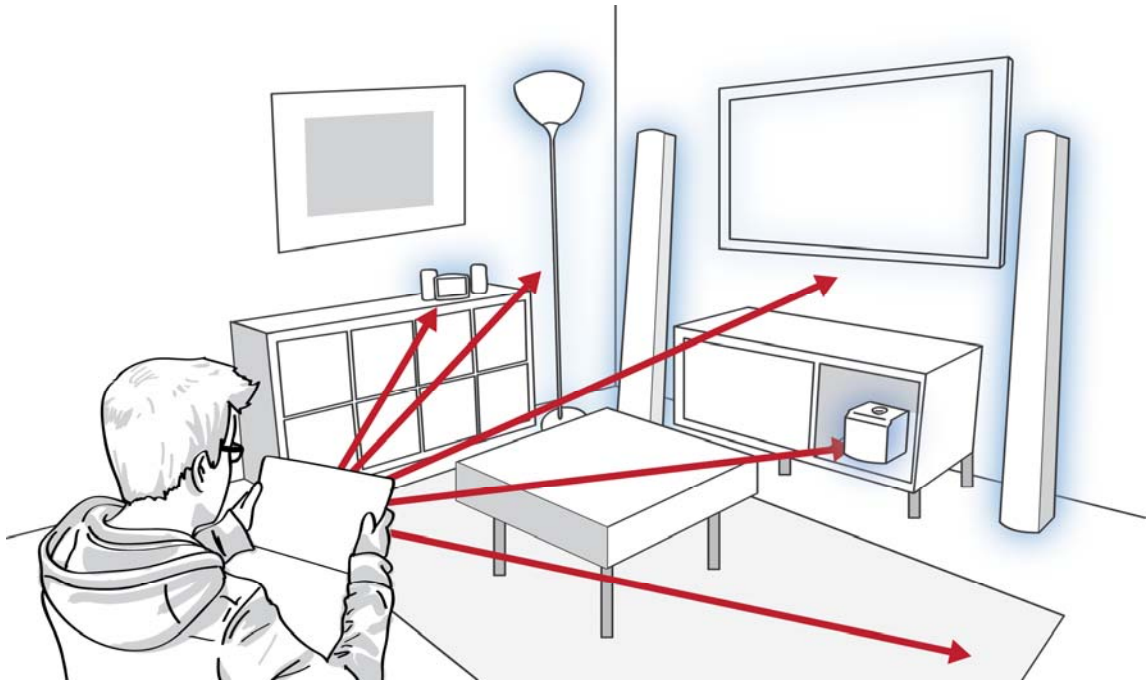


Figure 1.4 Living room portraying the challenges of our research, a large number of appliances arranged within a room. The person holds a tablet which serves as a reference point to these appliances in order to control them.

technology and interaction in the real world context to facilitate natural social practice. If we reconsider appliance control in ubicomp ecologies via a conventional mobile device through these notions, four problems emerge.

1. **It is difficult to know which appliances possess interactive capabilities.** When a person walks into a room, there is no way for them to know (1) the location of different interactive appliances; and (2) whether an appliance can be controlled.
2. **It is difficult to select an individual appliance within an ecology of devices.** As the number of devices in the ecology increase, it becomes increasingly difficult to select an individual appliance in order to control it.
3. **It is difficult to provide detailed information about an appliance.** Appliances typically contain a lot of information, such as indication of being on or off, the current task being performed, or how much battery is left. While typically a remote control should not necessarily show all information all the time, this information should be accessible or displayed when warranted.

4. **It is difficult to provide pertinent controls for appliances.** This problem is similar to revealing information about an appliance, however it is concerned specifically with how and when controls should be presented and focuses on enabling a user to act on an appliance as opposed to simply view their content.

It is our vision that we can work to integrate appliances into our ubicomp ecologies and enable more seamless interactions with them. Figure 1.4 summarizes our interpretation of the problem and vision by portraying a room with a large number of interactive appliances and an individual using a mobile device to control them. Our premise is that can take advantage of the spatial locations (e.g. distance and orientation) of these appliances as a way to expose information and controls. As it will be shown soon, this use of spatiality to interact with other devices pertains to an area of Human-Computer Interaction known as proxemic interaction, which leverages social protocols to enable users to perform seamless cross-device communication. These notions are accepted by the research community and thus lead us to believe we can apply these concepts to explore how different users in a household interact with appliances as a design challenge. Thus, we seek to explore this thread as an alternative way of interacting with an ecology of appliances, without claiming to replace current existing technologies such as traditional remote controls, or proving that this is a technology that people will adopt in the future. As such, the research provided in this thesis assumes that proxemic interactions leverage multiple devices and socially established protocols to create a seamless interaction between people and such devices. Consequently, my thesis is that

we can leverage mobile devices and socially established protocols, in this case, proxemics, as a way to seamlessly reveal information and controls for different appliances in a room.

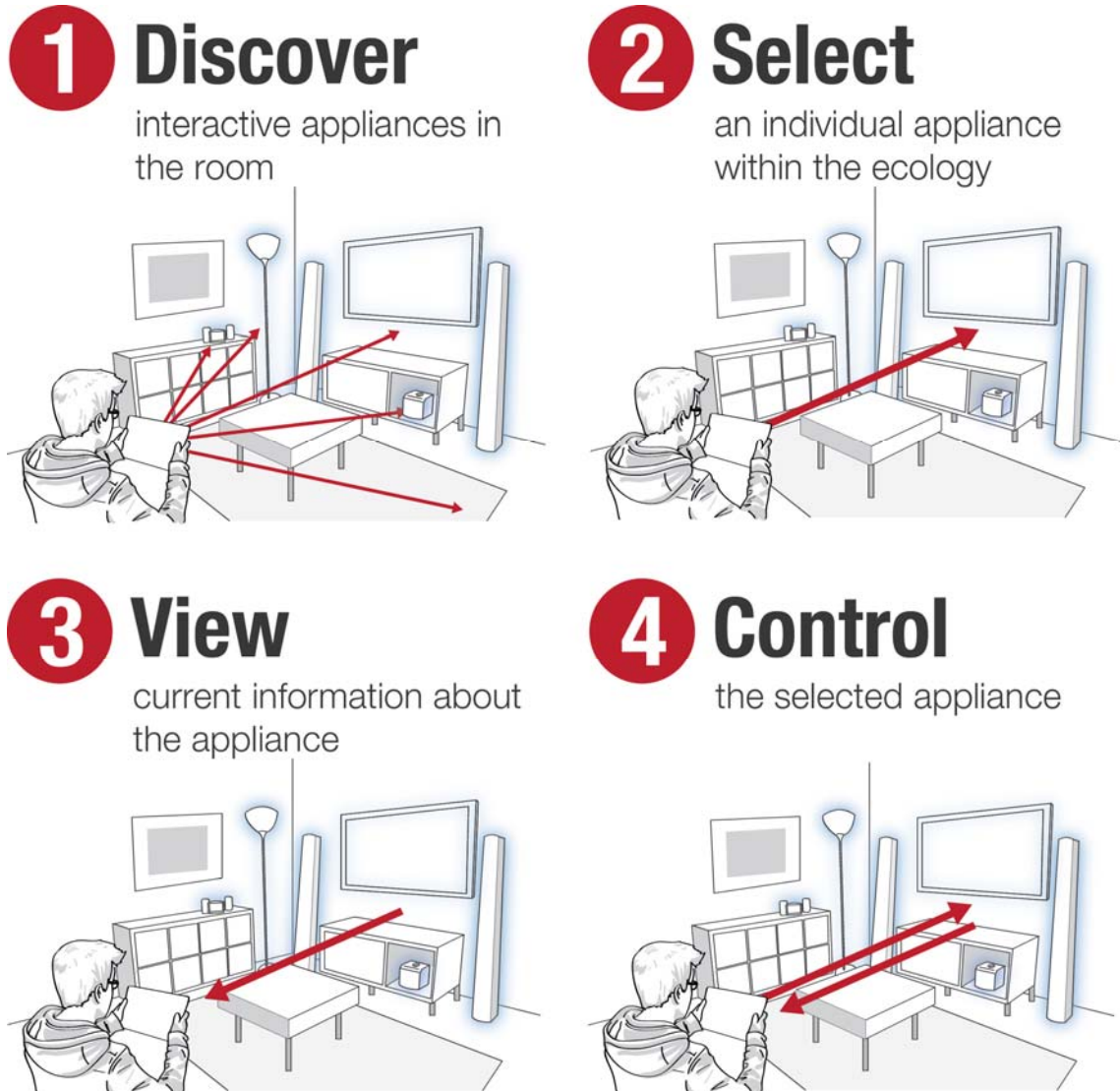


Figure 1.5 Summary of this thesis' goals

1.3 Approach and Research Objectives

In order to accomplish this seamless interaction through social protocols, this thesis will leverage current work in ubiquitous computing, in this case, proxemic interaction. As described in depth in Chapter 2, *proxemic interaction* refers to the use of spatial cues about people and devices (distance, orientation, movement, identity and location) to inform the design of interaction techniques in ubicomp ecologies (Marquardt, 2013). These concepts are directly based on the social theory

of proxemics (Hall, 1969), which explains human-to-human interaction and non-verbal communication. One application of proxemic interaction is the concept of *gradual engagement*, which establishes how devices can engage people into interacting with them (ibid). This is done by providing interaction opportunities as one approaches the device through (1) awareness information that acknowledges the person's presence, (2) providing opportunities to view information (such as progressive reveal as one approaches the device), and (3) enabling the person to engage in action. Current applications of gradual engagement have focused on large display interaction (Ballendat, Marquardt, & Greenberg, 2010; Vogel & Balakrishnan, 2004) or to support cross-device information exchange between digital surfaces (Marquardt, Ballendat, Boring, Greenberg, & Hinckley, 2012). Our main goal is

to adapt this notion of gradual engagement to satisfy interactions with ubicomp ecologies in a broader sense and enable control of appliances in the room.

To achieve this broad goal, we address the problems illustrated in the previous section through four sub-goals, summarized in Figure 1.5:

- 1. Discover interactive appliances within a room.** In order to achieve this goal, I will explore how we can use our mobile devices to find interactive appliances in the room. This is done in part by monitoring the spatial relationships between the interactive appliances and the mobile device through the use of proxemics and gradual engagement to reveal them on the mobile device. I will also apply visualization techniques that make use of these spatial references to provide awareness of appliance locations to individuals.
- 2. Select individual appliances.** To accomplish this goal, I will investigate how we can apply proxemic relationships so individuals can select and interact with a specific appliance within the ecology. This also means that a person will be able to transition from one appliance to another without

disrupting the content presented on the screen. Thus, I explore how our technologies can facilitate switching context between being engaged with a particular appliance to engaging with another one.

3. **View information about appliances.** Managing the display complexity of state information about appliances, such when to show details about the current task or specific details such as battery level, are often not considered when designing remote controls. To achieve this goal, our use of gradual engagement will aim to progressively reveal pertinent state information to individuals.
4. **Control appliances.** Beyond exploring state information, people have to be able to change the appliance's state and settings. By adapting gradual engagement to this context, I will try to relax the notion of state and controls with the purpose of increasing both flexibility and usability without compromise. To do this, we will explore how we can leverage an appliance's complexity to transition between different controls.

In order to achieve the aforementioned goals, I will explore the design space of appliances and determine mechanisms to adapt the gradual engagement design pattern to how we interact with appliances. I will then prototype a room environment capable of tracking the spatial relationships between a person's mobile device relative to the surrounding appliances. Given that our room prototype is experimental, we will use a mix of actual networked appliances and proxy placeholders that mimic (but do not actually implement) appliances. The mobile device will be able to connect to the appliances, show information about them and change their current state. All of this will be done using a combination of top-down and bottom-up design approaches.

1.4 Methodology

This thesis focuses on presenting the final outcomes of the design process. However, in order to realize our solution, we followed a research through design approach (Zimmerman, Forlizzi, & Evenson, 2007). This means that we explore different designs with the goal of proposing a solution to the problem and also providing a better understanding of the space. The design exploration took place through a combination of *top-down* (using knowledge to inform implementation) and *bottom-up* (creating prototypes to guide exploration and provide reflection) approaches. Throughout each of these stages, the work was demonstrated to several people (experts and non-experts) who visited the lab and offered insight as to different directions the work could take. This thesis' appendix outlines the process and retroactively describes the process in the context of research through design approaches. The design process incorporated sketches throughout the design process to explore multiple ideas, which Bill Buxton describes as getting the right design as opposed to getting a single design right (Buxton, 2010). As multiple ideas are explored, the ideas become more focused and serve to improve the current design rationale. This ideation process is highly tied to existing theory in Human-Computer Interaction, where we use the theories to articulate our design decisions and as vocabulary to describe them. Our combination of existing theories creates a new hybrid able to contextualize these ideas, which fall in line with Yvonne Rogers' (2004) explorations of generative theories. The sketches are selected and refined into software, and those implementations inform the design of a major prototype. The prototype served to demonstrate the concepts in action and show the feasibility of the adapted theory, along with our new design constructs. We encourage readers to examine the appendix to this thesis.

1.5 Research Context and Scope

Figure 1.6 shows our research scope. This thesis is primarily concerned with the design of proxemics-aware controls, which are a means to control multiple appliances within a ubicomp ecology. In turn, this is a specific application of proxemic interaction, which leverages the spatial relationships between people and devices to make use of interconnected devices within a ubicomp ecology. Ubicomp ecologies can be considered a subdomain of ubiquitous computing, which has the broader goal of integrating technologies into our everyday life. This follows from Human-Computer Interaction (HCI), the study of how to create technologies that meet people’s everyday needs.

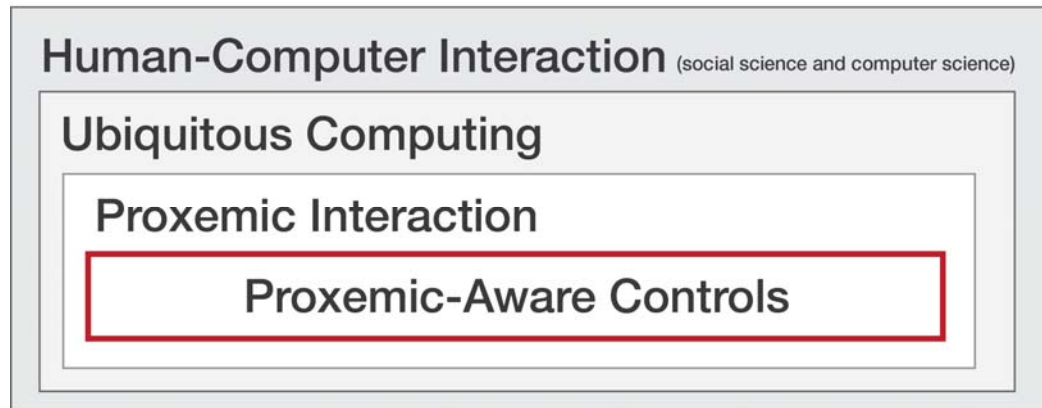


Figure 1.6 Research context and scope.

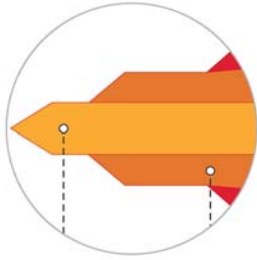
1.6 Thesis Structure

This thesis is comprised of six chapters, each of which shows the progression from the problem to the proposed solution.



Chapter 2 – Related Work

I present a history of remote controls. This is followed by prior research that concern interaction with appliances within HCI and how we can extend these techniques through the use of proxemic interaction.



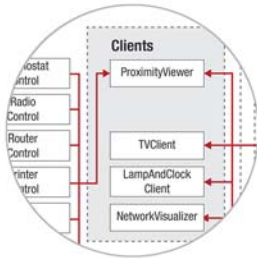
Chapter 3 – Conceptual Framework

I present a conceptual framework that describes the breadth of appliances that can be controlled and then structures proxemic interaction within it.



Chapter 4 - Proxemic Aware Controls in Action

I reveal the different appliance interfaces and operationalize the concepts from Chapter 3 through a prototype explained as a set of usage scenarios. These scenarios discuss the nuances of gradual engagement as an interaction approach.



Chapter 5 – System Details

I show the implementation details involved in our proxemics-aware controls system showcased in Chapter 4. This includes how the appliances are implemented, different presentation techniques examined and the overall system architecture, followed by limitations of the current work.

Chapter 6 - Conclusions

I summarize the accomplishments and contributions of this thesis and present several future directions.

Chapter 2. Background and Related Work

“Any technology that is going to have significant impact over the next 10 years is already at least 10 years old. That doesn't imply that the 10-year-old technologies we might draw from are mature or that we understand their implications; rather, just the basic concept is known, or knowable to those who care to look.”
- Bill Buxton *“The Long Nose of Innovation”* (Buxton, 2008)

This chapter provides background for this research through a literature review of approaches relevant to remotely controlling appliances. §2.1 describes traditional remote controls and summarizes their emergence and evolution into an everyday commodity. §2.2 explores how the proliferation of digital appliances in small spaces defines a *ubicom ecology*. Here, users now have the added burden of discovering and selecting those appliances they wish to view and control. We also describe various techniques for browsing available appliances in the ecology. Finally, §2.3 outlines the idea of gradually engaging with appliances via proxemic interaction, a theme developed further in Chapter 3 that will inform our use of ***socially acceptable protocols to enable seamless interaction with appliances.***

2.1 Remote Controls Yesterday and Today

A remote control is an electronic device capable of operating an appliance wirelessly¹. As a result, it is typically understood that these devices are portable in order to operate the appliances from a distance without the need to physically approach them.

Remote controls have a long history². The original intent for remote controls was for war purposes, in which the ideal scenario was to control attack machinery from a distance. The first form of remote control was invented by Nikola Tesla in 1898, a device called a Teleautomaton (Figure 2.1 a³) that was used to control a boat from a distance using a telegraph signal (Marincic, 1998). Everyday routine use of

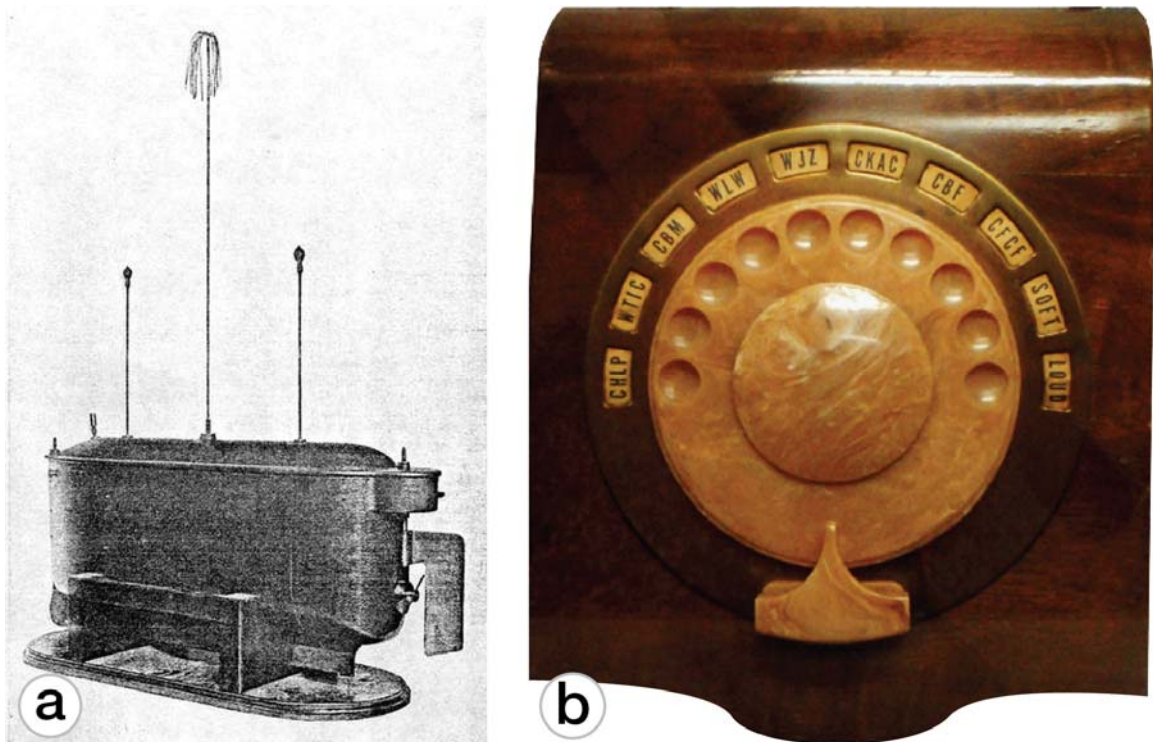


Figure 2.1 Some of the first remote control technologies: (a) Tesla's Teleautomaton and (b) Philco's Mystery Control, the first wireless remote control for a radio.

¹ Based on definitions from Merriam-Webster Dictionary and Wikipedia

² Based on <http://science.howstuffworks.com/innovation/repurposed-inventions/history-of-remote-control.htm>

³ Reproduced from <http://cyberneticzoo.com>

remotes for controlling a home appliance, in particular, a radio, appeared in 1939. It was known as the Mystery Control, and was created by a company called Philco. As shown in Figure 2.1 b⁴, the controller was a box with a dial that had to be carried with two hands. The box featured a dial similar to those in old dial telephones, and enabled people to select specific radio stations, switch the input between radio and vinyl discs, and change the volume or mute the radio through RF signals.

In 1950, the Zenith Radio Corporation introduced the first remote control for television, “Lazy Bones” (Figure 2.2 a⁵). As seen in the advertisement, the remote was coupled by a cable to the television, and its controls allowed a person to change channels and turn the TV on or off (Luplow & Taylor, 2012). Eventually, remote controls and devices came equipped with infrared communication, which was an important evolutionary step that untethered the connection between the remote and the appliance. From this moment on, different forms of remote controls proliferated.

As the number of devices that could be controlled increased, so did the number of remote controls, each with different functions and interfaces. A sampling is shown in (Figure 2.2 b) ⁶, which illustrates various remote controls for entertainment-type devices (e.g., television, music players). This led to various problems. Due to the sheer number of remotes and the inconsistent interfaces between them (e.g., see the different button configurations in Figure 2.2 b), most people used only their most basic functions (Nielsen, 2004). In addition, having multiple remotes meant that they could be easily misplaced (ibid).

In response, the first universal remote was created in 1985, which could be paired with, and thus control, multiple electronic devices (Goodson, McIntyre, & Rumbolt, 1987). The premise was that it could be taken from room to room, allowing people to use it to control most appliances without much effort. However,

⁴ <http://gizmodo.com/5857711/philco-mystery-control-the-worlds-first-wireless-remote>

⁵ Reproduced from <http://www.electronichouse.com>

⁶ Taken from <http://www.instructables.com>



Figure 2.2 Different remote control technologies: (a) shows Zenith LazyBones and (b) shows more modern remote controls including a universal remote control.

universal remotes did not supplant multiple dedicated remotes, largely because universal remotes were both difficult to configure (Nielsen, 2004) and also had a confusing array of controls (e.g., see the universal remote in Figure 2.2 b, far right).

The above complexity is still present today. Dedicated remote controls are still proliferating, in part due to this complexity and also because it is cheaper to manufacture a remote control rather than having dedicated controls on the actual appliance. As the diversity of appliances increase, ‘traditional’ universal remotes became less suitable as a generic solution. This may be why the focus for universal remotes to date has been for entertainment centers, in which multiple devices operate together in the same room and use similar controls, such as a home theater comprised of a TV, a sound system, video players, etc.

In 2002, Brad Myers suggested that mobile devices, rather than dedicated remotes, could be a convenient way to control everyday appliances (2002). His premise was that mobile devices would become commonplace in the future, and

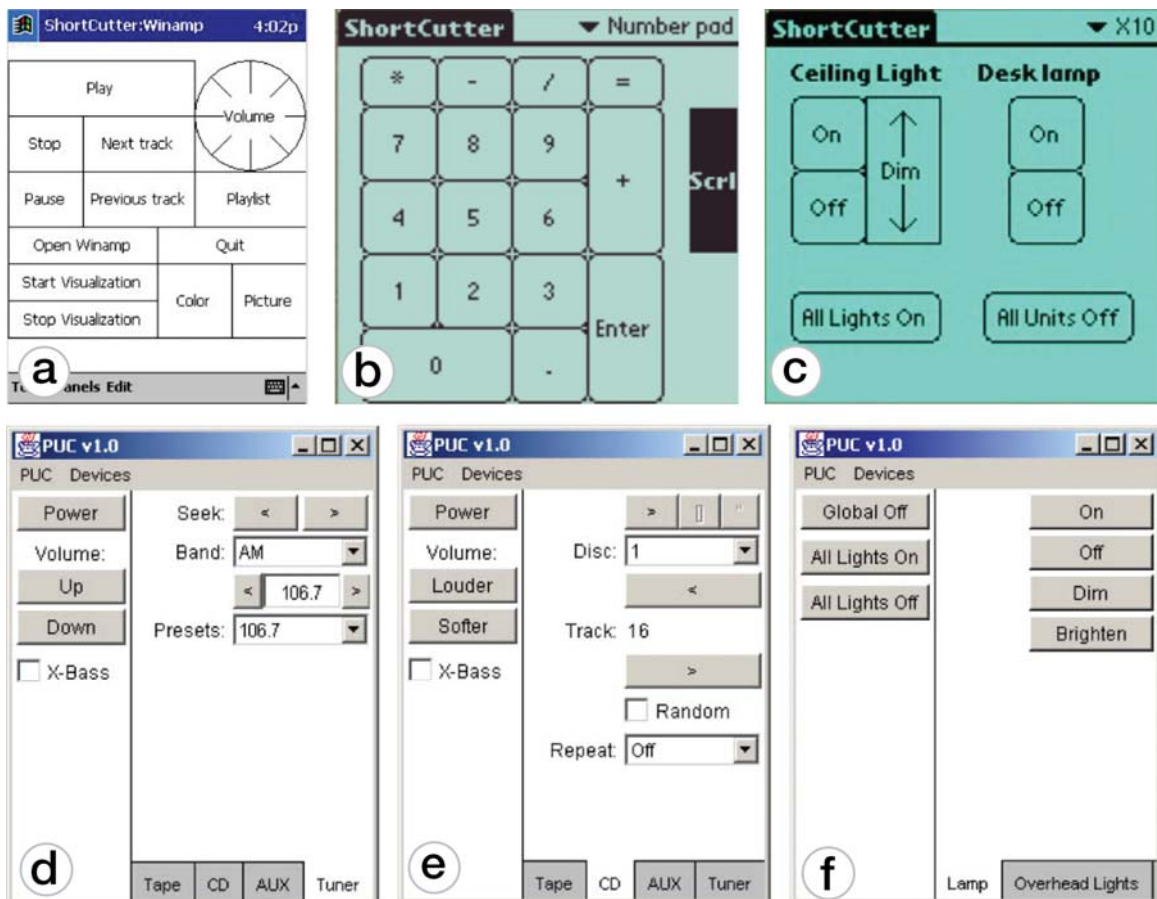


Figure 2.3 PDA for remote controls: (a), (b) and (c) show ShortCutter, an application meant to be used aside a keyboard, and (d), (e) and (f) show next iterations of it as a Personal Universal Controller, which generates interfaces on the device automatically. *Reproduced from (Myers, 2002).*



Figure 2.4 Popular mobile remote applications: **(a)** shows the iPhone Remote App, and **(b)** shows the Nest Thermostat along with its corresponding mobile app.

that their higher computing capability would allow for mutable graphical user interfaces that could adapt to different contexts. Figure 2.3 a, b, c provides examples of this running within the PDA ‘ShortCutter’ app, in which its graphical user interface changes to display controls for three different devices: a music controller (2.3a), a keyboard number pad (2.3b), and a light controller (2.3c). As device configuration is a potential challenge for users, Nichols et al. later proposed ways in which interfaces could be encoded and transferred between appliances and mobile devices as ad-hoc XML descriptions (Nichols et al., 2002). For example, Figure 2.3 d, e, f⁵ illustrates 3 interfaces running on the Personal Universal Controller (PUC) application. Each is configured by its XML description, where each is built on standard graphical user interface components (e.g., buttons, checkboxes, drop-down menus).

Today, as anticipated by Myers (2002), mobile devices have become ubiquitous. Indeed, many mobile devices now offer mechanisms to connect and control particular digital appliances or software running on a computer. A popular example is the iPhone Remote App, which allows a person to remotely control the Apple iTunes music player or Apple TV. Figure 2.4 a⁷ illustrates iPhone’s mobile interface reflecting the current iTunes playlist (also seen on the larger computer

⁷ Reproduced from <http://www.engadget.com/2008/07/10/apples-remote-control-application-for-itunes-and-apple-tv/>

screen). The iPhone Remote App, amongst other things, shows which song is currently playing, and allows people to select specific songs, navigate through playlists, and change the volume. Dedicated apps as exemplified by the iPhone Remote App can have carefully crafted interfaces to provide a positive remote control experience. Another example, mentioned in Chapter 1, is the Nest thermostat. Nest thermostats can be accessed and controlled through a website, and also through an application that can be installed on a mobile device capable of remotely changing different settings. One of its screens is shown on Figure 2.4 b⁸.

Home automation systems comprising a range of products triggered the emergence of remote controls that could manage multiple home systems. The original versions were typically non-mobile and tethered, usually comprising a console at a fixed location somewhere in the home. For example, Plaisant et al. designed a touch-screen console for scheduling a variety of home-control devices, where they envisioned the console mounted at some central location in the home (Plaisant & Shneiderman, 1992). More recently, wireless remotes implemented as smart phone and/or tablet apps have been used for home automation. For example, SmartThings⁹ consists of four sets of devices, shown in Figure 2.5¹⁰: (a) a central hub that interconnects all the compatible devices; (b) sensors such as motion sensors and multi sensors capable of sensing angle changes, movement, vibration and temperature; (c) compatible appliances such as an electronic door lock; and (d, e, f) a mobile device interface that offers – via its various screens – the user access and control of the home automation system. For example, in order to incorporate a sensor or appliance as part of the interface, a user can manually add the appliance (Figure 2.5 d ‘Connect New Device’), or wait until the hub recognizes and adds new devices to the list of ‘Things’ shown as displayed in Figure 2.5 d. In order to control an individual appliance or to apply a setting, the user can

⁸ Reproduced from <http://www.slashgear.com/nest-thermostat-gets-insteon-smart-home-integration-12320277/>

⁹ <http://smarththings.com>

¹⁰ Reproduced from <http://smarththings.com>

select it from a list. Figure 2.5 e shows a grid with all the interactive devices that are currently interconnected, such as the lamp in the second row and second column. The settings for the appliances can range from explicit changes in state, such as turning on or off, or events, such as turning on when a sensor is triggered, or at a certain time. Finally, one can see an activity feed, similar to a usage history, which displays on-going activities of all connected appliances and their current state (Figure 2.5 f). For example, the most recent activity at 7.27 am, shown at the top of Figure 2.5 f, states that the ‘Kitchen light is switched on’.

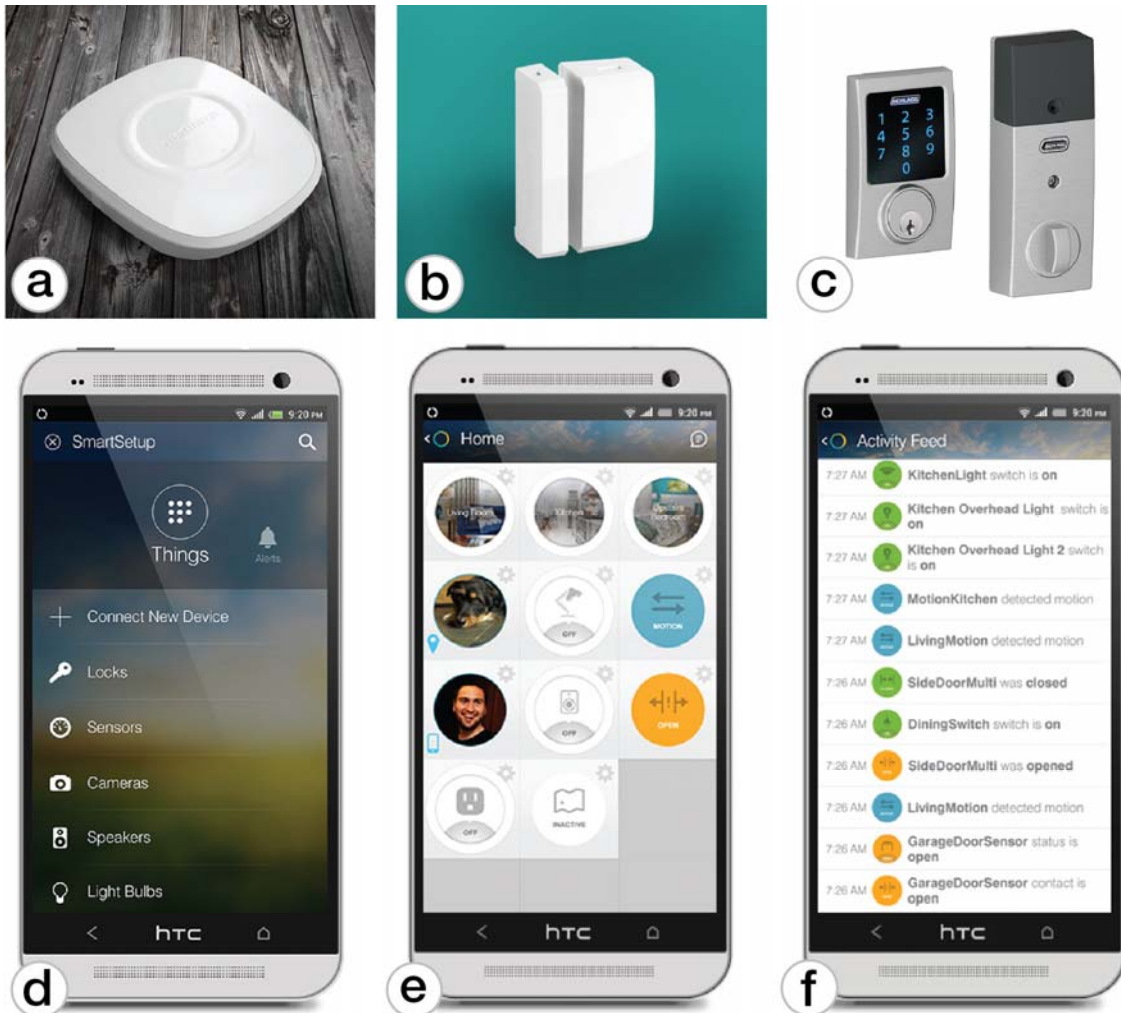


Figure 2.5 SmartThings: (a) the central hub (b) a multi sensor, (c) a door lock. The phone interfaces for SmartThings show: (d) the menu to configure a new device, (e) a list of connected appliances and (f) the activity feed of the appliances.

2.2 Appliances in Ubicomp Ecologies

The previous section broadly introduced the evolution of remote controls. This section reconsiders remote controls from a more contemporary perspective. In particular, it considers how a larger number of digitally controllable appliances co-exist within a small ecosystem called a *ubicomp ecology*, and on how a person actually discovers, selects and controls these multiple appliances within that ecology.

2.2.1 Ubicomp Ecologies

Digitally controllable appliances are now proliferating to the point that a room, office, home or other small contained spaces can be considered a technological ecosystem. Marquardt (2013) describes such ecosystems as *ubiquitous computing ecologies* populated by people and devices. As illustrated in Figure 2.6, people have

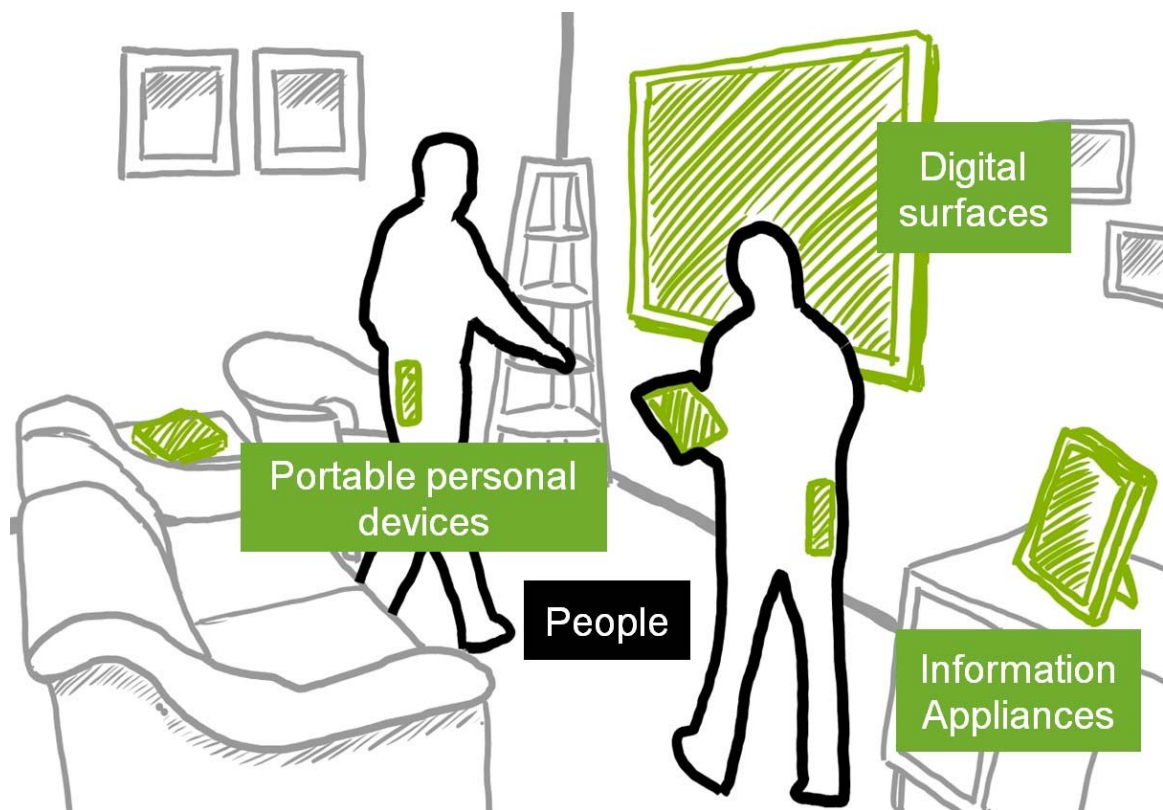


Figure 2.6 A sketch outlining ubiquitous computing ecologies. Here we can see a space with people and devices (digital surfaces, mobile devices and information appliances).
Reproduced from (Marquardt, 2013).

the opportunity to interact with multiple devices in a room. These can range from digital surfaces such as large displays or tabletops, to portable personal devices such as mobile phones, tablets and camera, and to various digital appliances, such as a digital picture frame. As a ubiquitous ecology, the devices contained within it are expected to be able to communicate with one another, making cross-device interaction possible.

Generally, a key goal in ubiquitous computing is to enable seamless interactions between people and the ecology as a whole. People should, of course, be able to discover and leverage the capabilities of the devices contained within it, including the ability to use multiple devices in concert. There are also different relationships that can occur in such ecosystems. One relation concerns person to device interactions, in which the device recognizes and reacts to a nearby person. An example is a large display that reacts to people approaching it, and tailors its content accordingly by revealing more information as one gets closer (Ballendat et al., 2010). Another example is the Microsoft Xbox Kinect¹¹. With the Kinect, games displayed on the screen react to the movement of people. Another relationship is device to device interactions, which allows nearby devices to recognize each other. An example is a mobile device that points to the contents of a large display and allows control of that display via an augmented reality approach (Boring, Baur, Butz, Gustafson, & Baudisch, 2010). Yet another relationship considers one device as the primary means to control another device, such as video game controllers.

2.2.2 Ubiquitous Appliances

UbiComp ecologies are very broad in concept. They can include quite complex systems that comprise a broad set of functions, and that have rich input and output mechanisms. Our particular interest is narrower: we consider how a person can remotely control simpler devices – which we call *ubiquitous appliances* - within these ubiquitous ecologies.

¹¹ <http://www.xbox.com/en-CA/Kinect>

What do we mean by ubiquitous appliances? The classic definition of a conventional appliance is ‘a device or piece of equipment designed to perform a specific task, typically a domestic one’¹². A ubiquitous appliance can be defined as ‘a *digitally controllable* device or piece of equipment designed to perform a specific task, typically a domestic one’. Yet this is too extreme a definition. Digital devices can range from simple highly task-oriented single-purpose devices at one end, and general purpose computational devices at the other. Our view of ubiquitous appliances is somewhat more relaxed, as they can perhaps go beyond a specific task. While they are generally task-specific, we accept that some appliances can have reasonable complexity and functionality (e.g., an entertainment center). Another issue is that traditional appliances are seen as ‘equipment’, which implicitly means they have a physical form. While this is true of many digital appliances (e.g., a computer-controlled lamp, a router), we also consider devices that comprise a physical interface to a larger underlying and invisible infrastructure (e.g., a thermostat to a heating system), or even virtual devices (e.g. integrated home audio system with no apparent physical form). To further complicate matters, digital appliances can differ greatly in their input and output capabilities, ranging from those with no external controls or with displays that must be operated externally (i.e. with a limited button pad), such as a home alarm system, to those with complex controls and displays (e.g., a digital picture frame). Even so, we generally consider ubiquitous appliances to have fairly limited input/output capabilities due to factors such as form factor, cost and comfort. These and other themes will be discussed further in Chapter 3.

2.2.3 Discovering and Selecting Appliances in a Ubicomp Ecology

A ubicomp ecology can become very complex as the number of digitally capable appliances increases. As a result and as stated in Chapter §1.2, it can be difficult to:

¹² http://www.oxforddictionaries.com/definition/american_english/appliance

- (1) discover interactive appliances;
- (2) select an individual appliance out of the ones in the ecology;
- (3) view relevant information about that appliance, such as its current state; and
- (4) control that appliance to change its state.

The first two steps are particularly important for remote control of ubiquitous appliances. First the remote control has to discover, connect and/or create an association with the appliance (either automatically or perhaps under control of the user). Then the user has to select the desired appliance in order to view and control it. Thus to be effective, remote controls have to dynamically discover and connect with nearby appliances, and the interface must make it easy for the user to see and choose between appliances.

Earlier work in mobile interactions more generally refers to this process as *physical browsing*: a means to discover interactive devices and retrieve their corresponding user interfaces. Physical browsing techniques are clustered into three categories (Välkkynen & Tuomisto, 2005). *Scanning* is done when a mobile device somehow acquires a list or other visual representation of all devices within the ecosystem and enables individuals to select the desired device. *Touching* refers to a mobile device making physical contact with the desired device to associate them, such as by tapping. *Pointing* at the desired device (e.g., with one's mobile phone or tablet) establishes an association at a distance. The next three subsections reviews each of these categories in detail, in the context of associating remote controls to appliances.



Figure 2.7. Scanning shows a list of connected devices. Here we can see (a) a list of devices from physical browsing research; (b) another interpretation by Yoon et al; and (c) Huddle, showing how users can select input and output devices for digital content. Reproduced from (Välkkynen & Tuomisto, 2005), (Yoon et al., 2007) and (Nichols et al., 2006) (annotated).

2.2.3.1. Scanning

Scanning covers the situation in which a remote control visually displays all appliances it knows about, and then allows the user to select particular devices to either connect with them or interact with them. It assumes that appliances have an

identity that can be detected by the user's mobile device through some discovery mechanism. An example discovery mechanism is the Bluetooth wireless protocol, in which all devices with this capability are listed and one can then pair the devices and control them together.

Some remote controls display appliances via a simple list. For example, Figure 2.7 (a-b) illustrates two such examples: as a linear list in 2.7 a, and as separate hierarchical tabs in 2.7 b. Appliances can also be displayed graphically, such as the icons in 2.7 c which further allows the user to interconnect multiple appliances, such as splitting a video into a television for visual output and audio to the home theater for audio output (e.g., streaming sound from one appliance to another).

As the number of appliances increase, it becomes increasingly difficult to navigate through such lists, as it results in cognitive overload (Rukzio et al., 2006). Referring back to Figure 2.7 a and b (Välkkynen & Tuomisto, 2005; Yoon, Kim, & Woo, 2007), we can see that these lists are straight forward when there are few devices, but if there are more interconnected devices, then identifying individual appliances can be difficult. Huddle demonstrated an extension to this approach through aggregation of appliances that perform joint tasks (Nichols, Rothrock, Chau, & Myers, 2006). However, as shown in Figure 2.7 c, the focus here is on distributing the flow of content to then generate an aggregate control automatically; the figure shows a DVD movie splitting its content into video for the television and audio for the sound system. Once the content was distributed, the system would generate the aggregate interface for these devices. This grouping of devices shows a mechanism to potentially reduce the number of items, particularly

when they perform joint tasks. This aggregation also simplifies control mechanisms (e.g. the redundant volume controls for the TV and for the audio system are merged into one).

Another approach represents devices by their spatial topography. Consider Tani et. al.'s notion of object oriented video (1992). In object oriented video (Figure 2.8 a), live video images of the devices being controlled are used to directly manipulate the device. For example, clicking on a knob portrayed in the video controls the

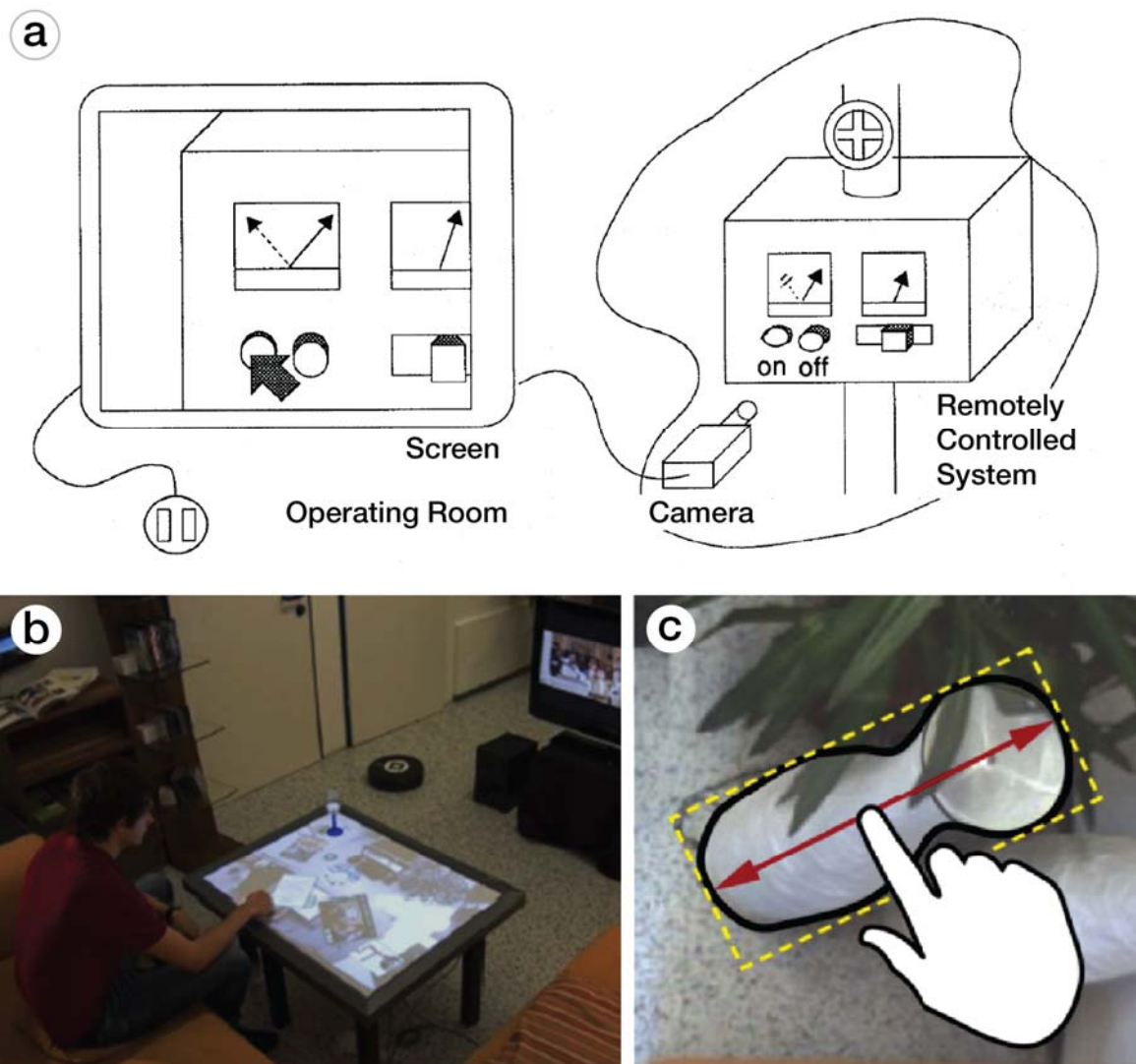


Figure 2.8. Scanning through video feeds showing: Object Oriented Video (a) showing a power plant controlled from a distance using a computer; and CRISTAL (b and c) the overall interface and an example of how a light is controlled and shows feedback live.

Reproduced from (Tani et al., 1992) and (Seifried et al., 2009) respectively.

physical knob at a distance. This method was originally considered as a means to operate devices located in potentially unsafe areas (such as a power plant). An important benefit is that one can visually see changes that occur in the distant devices being controlled through the camera image. CRISTAL (Figure 2.8 b and c) applies this concept to ubicomp ecologies. It displays a bird's-eye view of a room on the display (a tabletop surface) that allows people to directly touch the appliance image in order to interact with it (Seifried et al., 2009). Figure 2.8 b further

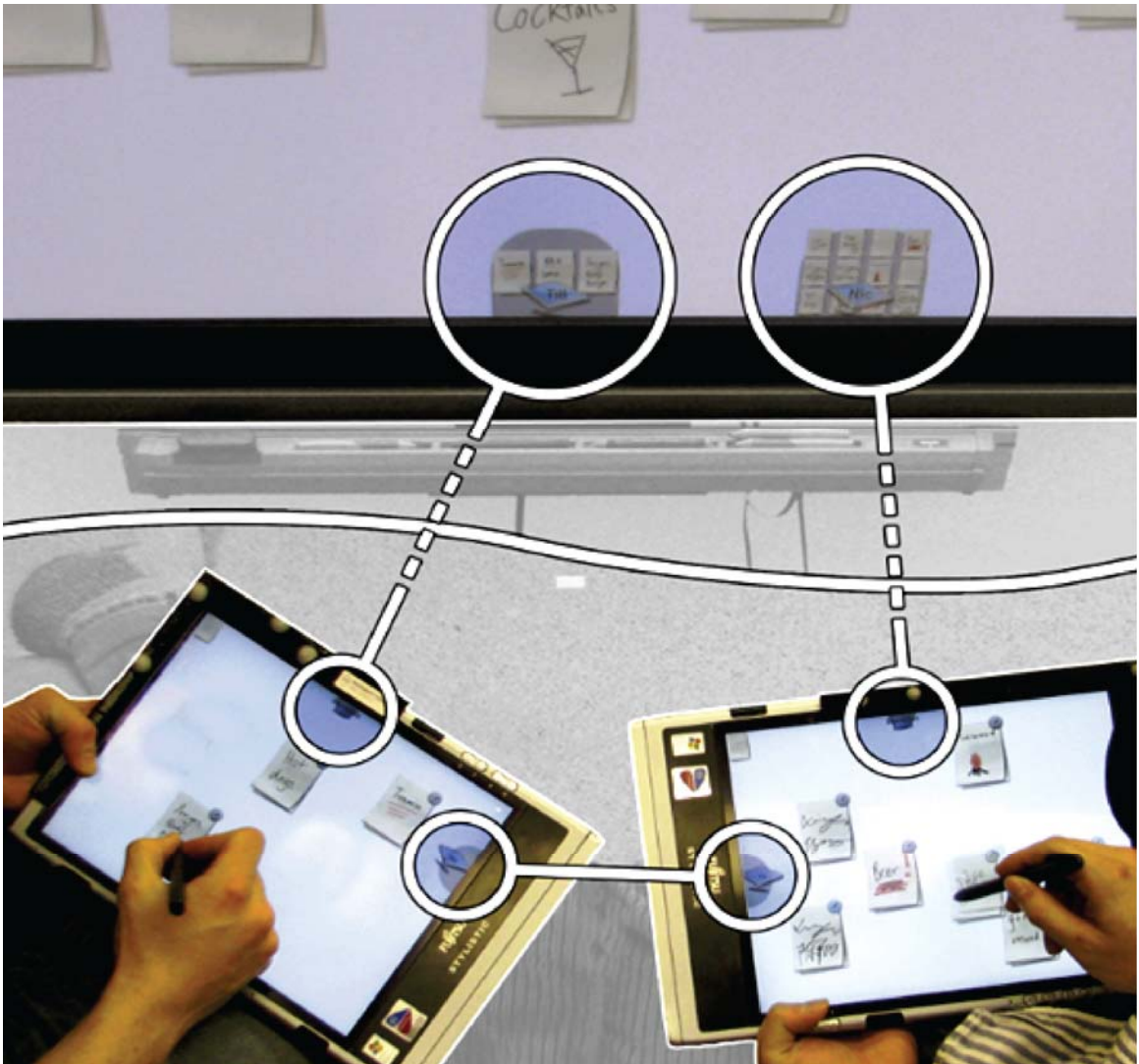


Figure 2.9. Spatial Referencing for discovery of devices: here we see two nearby tablets and a large display showing their spatial relationships between each other.
Reproduced from (Marquardt, 2013).

illustrates this, people can touch the image of a lamp and perform a drag gesture to change the illumination (Figure 2.8 c).

While video is quite literal, other forms of showing spatial topography exist. For example, the Relate Gateways system shows relative positions of appliances as icons overlaid on top of a tablet interface (Gellersen et al., 2009). This idea was further extended by Marquardt et al. (2012) through dynamic updates of the spatial references as people move, and through the ability of revealing content that could be shared. As shown in Figure 2.9, people working on their tablets can become aware of different devices that are currently interconnected, and their icons are positioned to correspond to the orientation between the tablet and the connected device. Consequently, it helps to better identify the devices, as opposed to requiring a user to tell them apart in a list. Furthermore, this interaction facilitates information transfer, as users can exchange information from their tablet to connected devices, such as dragging a file to the printer in order to print.

2.2.3.2. Touching

Touching one device (or bring the two devices in very close proximity) is another way of triggering an association. The premise is that touching two objects to associate them is easily understood – and usually easy to perform - by people. It drastically reduces accidental selections. Indeed, Rukzio et al. argues that people



Figure 2.10. Physical association proposed by Want et al. Here we see a person (a) tapping a printer to print a file, and (b) tapping a poster to reveal more information.

Reproduced from (Want et al., 1999)

who are already standing prefer physically approaching objects, as opposed to choosing other forms of selection mentioned in this section (2006).

RFID¹³ tags are the most common way to implement touch. For example, Want et. al. equipped a tablet device with an RFID reader, and various objects with tags (1999). When a user tapped a tagged object with the tablet, actions specific to that object would be triggered. For example, in Figure 2.10 a, the person taps a printer with the tablet to print the file currently opened on that tablet. Figure 2.10 b is similar, except that tapping the poster retrieves further information about the poster's content, see also (Välkkynen & Tuomisto, 2005). Another way in which touching associations are created is through bumping two devices. Unlike the use of RFID, bumping requires devices to be equipped with accelerometers. By examining which accelerometers were in motion at the same time, a central system could interpret whether these two devices were bumping. This was introduced by Hinckley et al. as synchronous gestures (2003) and later commercialized into Bump¹⁴ as a system that enabled connections between mobile phones from multiple operating systems.

While touching a device to retrieve content makes selection easy, knowing which devices are tagged can be problematic unless tags or their location are somehow visibly marked. Because one has to essentially 'touch' the object, there is no easy way to preview the scene to see what objects are in the ecology.

2.2.3.3. Pointing

Pointing a mobile device towards the object they wish to interact with is appropriate when the two are distant (e.g., a few meters) from each other. Many technologies enable pointing, such as infrared (Beigl, 1999; Chen et al., 2013; Myers, et al., 2001; Swindells et al., 2002; Välkkynen & Tuomisto, 2005), computer vision (Kohtake, et al., 2001), or light sensing (Schmidt et al. , 2012).

¹³ RFID stands for Radio Frequency Identification

¹⁴ [http://en.wikipedia.org/wiki/Bump_\(application\)](http://en.wikipedia.org/wiki/Bump_(application)), discontinued January 31, 2014

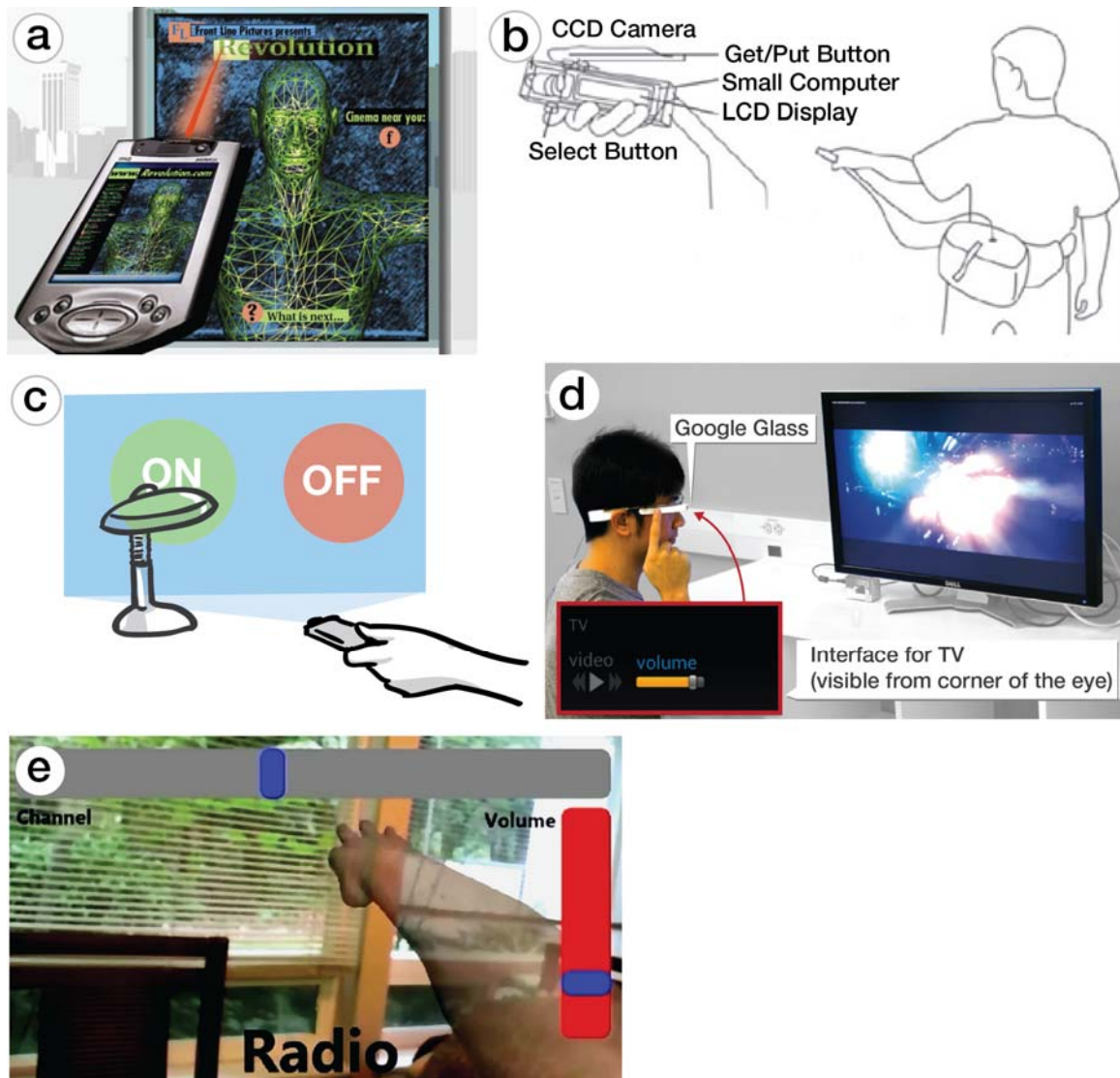


Figure 2.11. Work in pointing techniques. It shows: (a) Physical Browsing Research; (b) InfoPoint's pointing device; (c) PIControl turning on a lamp; (d) peripheral context menu on Google Glass; (e) using Digits to control a radio with augmented reality. Figures reproduced from (Välkkynen & Tuomisto, 2005), (Kohtake, et al., 2001), (Schmidt et al., 2012) (sketch), (Chen et al., 2013)(annotated), (Kim et al., 2012).

An advantage of pointing is that the mobile device can display information about the target as soon as it is aligned with it. For example, Figure 2.11 a illustrates a PDA interface that shows information on its screen in this manner. Other interesting variations to this approach exist. InfoPoint (Figure 2.11 b) enables the retrieval of information from one appliance in order to push it into another device (Kohtake et al., 2001). For example, one can point to a camera and press the 'Get'

button to fetch an image, and then point to a printer and press the ‘Put’ button to print that image. PICOntrol (Figure 2.11 c) leverages a mobile projector to reveal an interface with control options overlaid on top of the physical appliance (Schmidt et al., 2012), such as a lamp with controls showing the on and off state. By pointing the desired state on top of the appliance and pressing a button, the action would be triggered. Figure 2.11 c shows a sketch of how one can turn a lamp on by projecting the ‘on’ state on top of it. Chen et al. (Figure 2.11 d) use a head mounted display (Google Glass¹⁵) to reveal context menus for appliances pointed to by the head (Chen et al., 2013).

Pointing can be augmented by other richer gestures. In the augmented reality system illustrated in Figure 2.11 e, the user uses finger gestures to perform fine-grained parameter adjustments on a radio, such as a turning gesture to change the volume of a distant radio (Kim et al., 2012).

Rukzio et al. (2006) argue that pointing is the technique of choice when people are sitting, as often times they do not want to stand up to perform an interaction. However, pointing can be problematic with distant targets. Small movements of the hand can drastically change the direction of the pointing, leading to difficult selection and – if targets are close together – false positives. This compromises scalability.

2.2.3.4. Summary

In summary, scanning, touching and pointing are all known methods that help a user associate a remote control with an appliance in a ubicomp ecology. Each method has both strengths and weaknesses. Scanning provides great ways to encourage discovery of devices, yet there has been little focus on how one could integrate controlling interfaces. Touching enables easy selection at the expense of discovery. Pointing is a technique suitable for comfort, as people do not need to

¹⁵ <https://www.google.com/glass>

physically approach targets, but leads to problems of scalability and selection ambiguity (false positives) when there are too many objects in the scene.

This thesis proposes a fourth method that combines the best of each: gradual engagement via proxemic interaction. The idea is that one can orient one's remote around the room (pointing) to see what devices are available (scanning). The user can gradually engage with a chosen device by approaching it, and as a result, information about that device is revealed progressively (a variant of pointing and touching). This technique aims to **leverage socially established protocols as a way to seamlessly reveal information and controls for different appliances in a room**. Proxemic interaction and gradual engagement are introduced in the next section, and contextualized further in subsequent chapters.

2.3 Proxemic Interaction and Gradual Engagement

American anthropologist Edward Hall devised the theory of proxemics to explain his observations of how nearby people interacted with each other (Hall, 1969). This theory asserts that spatiality is a key form of non-verbal communication and that spatial relationships such as distance and orientation between individuals expose how people tend to equate physical distance to social distance.



Figure 2.12. Example of proxemic distances showing: (a) intimate; (b) personal; (c) social; and (d) public zones.
Reproduced from Hall (1969)

Hall defined the concept of proxemic zones, Figure 2.12 highlights some of these spatial relationships. While the distances defining a proxemic zone can vary according to the culture, the zones generally preserve the same meaning. Therefore, the distances are only used as references and are not necessarily clear-cut. The **intimate space** (0-50 cm) takes place when individuals are in a close relationship or an argument (shown in Figure 2.12 a), it is a space normally entered with permission. The **personal space** (0.5-1.2 m), shown in Figure 2.12 b, is the relationship that one typically uses with friends and family, in which people can still touch each other and speak at a lower volume. The **social space** (1.2-3.5 m), Figure 2.12 c, is a distancing that entails more formality, physical contact is harder to perform and people are required to speak louder to each other. Finally, **public space** (> 3.5m) is a space in which vision is the primary sensory input and requires people to speak louder in order to address others, such as the speaker shown in Figure 2.12 d.

Proxemic theory was later re-interpreted within Human-Computer Interaction research as *proxemic interaction*. This was first done as means to interact with large ambient displays that revealed more information as people approached them (Vogel & Balakrishnan, 2004), and then as a way for people to interact with increasingly large ubicomp ecologies. These ecologies were comprised of people, devices, and fixed and semi-fixed features (Greenberg, Marquardt, Ballendat, Diaz-Marino, & Wang, 2011).

Ballendat et al. operationalized proxemics in ubicomp by looking at five different variables: distance, orientation, identity, movement and location (Ballendat et al., 2010a). This means that by looking at these variables, a system could predict people's intent. For instance, a person facing a large display and walking towards it would make the system interpret that the person intends to interact with it. The display then proceeds to reveal more information and invite the user to come closer, as well as reveals more interaction possibilities.

One way in which proxemic interaction are applied within cross-device interaction is through a design pattern known as Gradual Engagement (Marquardt et al., 2012). Marquardt and colleagues devised this pattern to generalize common applications of proxemics within the field of Human-Computer Interaction. The pattern presents different stages of interaction between people and devices taking place in a determined order as a process of people approaching a particular device. These stages could be transitioned in two ways: through *discrete stages*, in which crossing a particular distance threshold produces a discrete change on the interface or *continuous*, where the interaction produces smooth changes and transitions as a function of distance. The stages of interaction in the context of cross-device interaction, as shown in Figure 2.13, are:

- (1) awareness of interaction opportunities with devices,
- (2) reveal of information, and
- (3) engaging in action through information transfer.

To illustrate, Figure 2.14⁷ shows a person with a digital camera gradually engaging with a large display. From afar, the display shows an icon of the camera that updates the camera's relative position in real time (Figure 2.14 a). This indicates that it has made an associating with the camera. As the person moves closer, he is able to see pictures that are contained in his camera (Figure 2.14 b). Finally, when near the large display, the person is able to drag out the images in the camera using direct touch (Figure 2.14 c). Marquardt et al. provide other examples of gradual engagement, such as showing the proxemic interplay between two tablets and a



Figure 2.13. Sequential stages presented in the gradual engagement design pattern for cross-device interaction.

Reproduced from (Marquardt, 2013)

large display and how interaction mechanisms can ease information transfer between these devices (2012).

Overall, proxemic interaction and the gradual engagement pattern provide a mechanism to interact with a large ecology comprised of appliances in a home environment. The next chapter elaborates this and contextualizes it for remote control of appliances. These ideas can help design interfaces that blur the seams between discovery, selection, viewing of information and controls to improve the user experience.

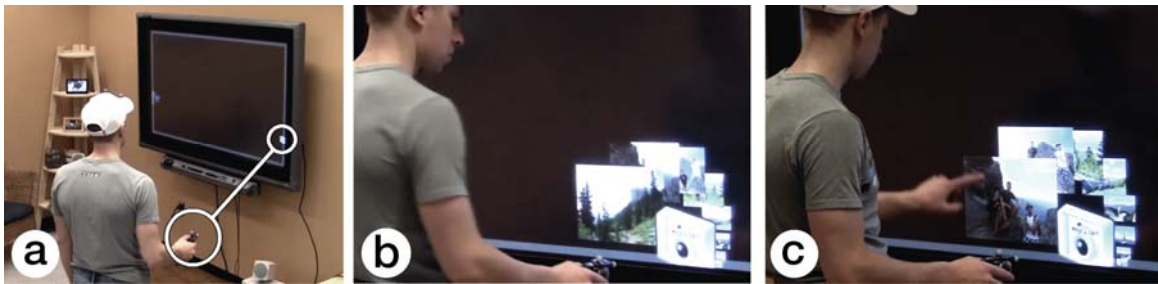


Figure 2.14. Example of the gradual engagement design pattern: (a) first the person approaches the display, the display acknowledges presence of the camera; (b) then the camera icon reveals the contents of the camera and (c) the person is able to drag files into the large display through direct touch.

Reproduced from (Marquardt, 2013)

Chapter 3. Conceptual Framework

The previous chapter described the evolution of remote controls over time, and the different ways in which one can interact with appliances within a ubicomp ecology. It also introduced the idea of proxemic interaction as a potential approach to better enable appliance discovery and interaction in such ecologies. This chapter presents a conceptual framework that aims to help structure the design process for proxemic-aware controls, which is summarized in Figure 3.1. This conceptual framework in turn guides the design of the prototypes presented in the next chapter by describing two main components: ecologies of appliances and proxemic interaction. Considering an ecology of appliances showcases the breadth of controllable appliances and informs considerations for system design (§3.1). Operationalizing proxemic interaction for remote control establishes the rules for interaction and aims to inform the interaction and interface design (§3.2). Figure 3.1 illustrates the different dimensions taking place in the framework, which will be explained throughout the rest of this chapter.

3.1 An Ecology of Appliances

As discussed in Chapter 2, a ubicomp ecology is comprised of people and devices, where the devices can vary from systems with large computational general purpose capabilities such as digital tabletops, to personal devices such as tablets and mobile phones, to appliances, such as a radio or a printer. Having insight about the breadth of appliances that can be controlled, along with their attributes can inform system considerations. It can help designers determine relevant controls to

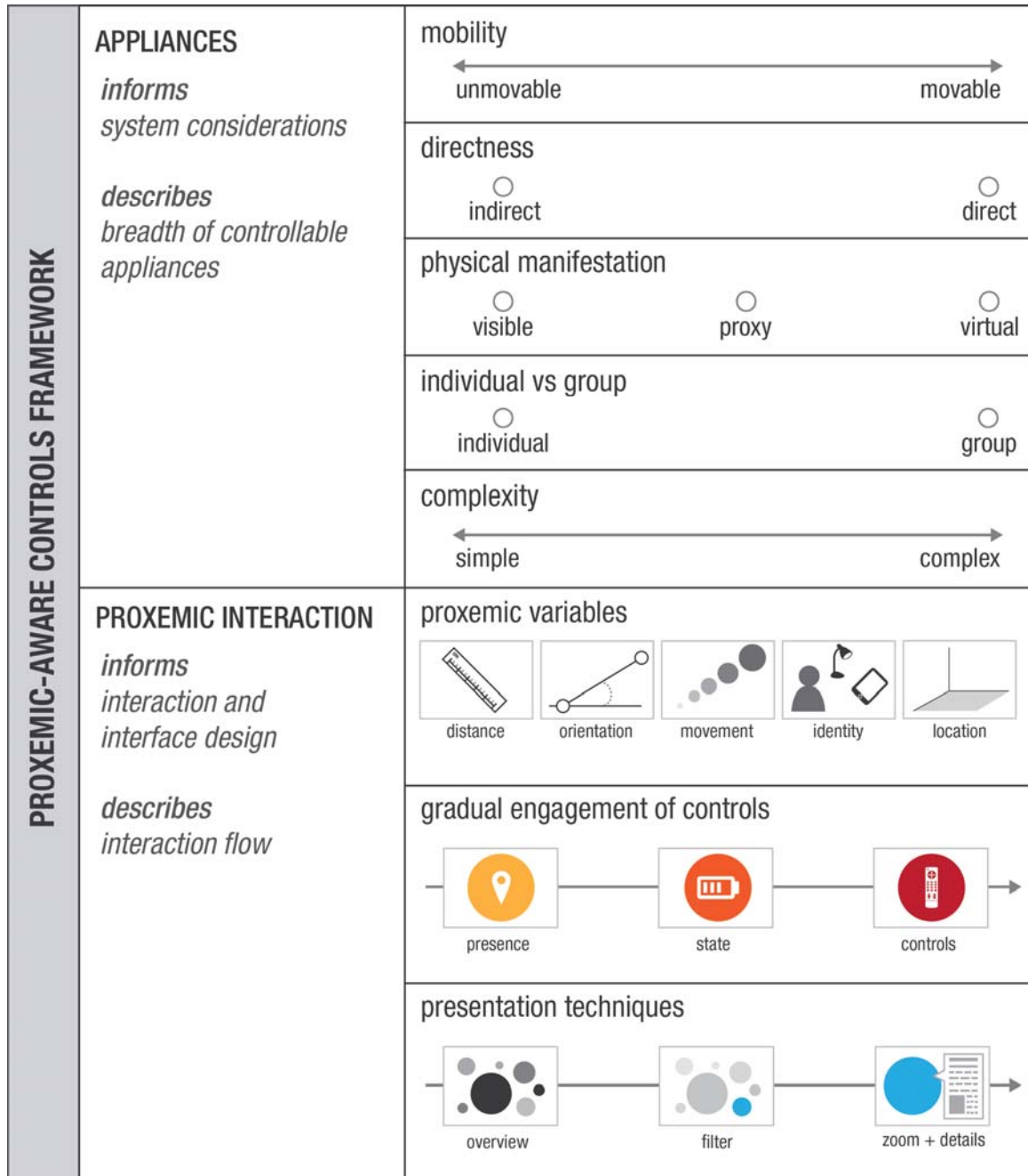


Figure 3.1 Diagram summarizing the Proxemic-Aware Controls Framework.

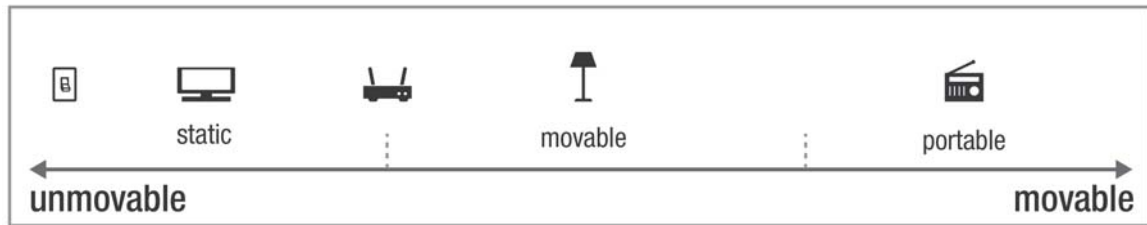


Figure 3.2 A Spectrum categorizing appliances according to their mobility.

extent of necessary tracking technologies). The next subsections describe different appliance attributes encompassed in this framework.

3.1.1 Appliance Mobility

The first appliance attribute is its mobility. Hall described how objects affect proxemics by placing them in two different categories: *fixed features* and *semi-fixed features* (1969). *Fixed features* account for elements that are embedded into the environment and unmovable, such as a building, a wall or a window. Proxemics are affected by fixed features such as these because they are often considered boundaries that further separate people. *Semi-fixed features* are elements that can be moved, such as furniture. Arrangement of furniture also affects proxemics. For example, a small inward-facing circle encourages social interaction, while the same chairs facing away from each other discourages it.

While Hall's observations focused on elements comprising a built environment, Marquardt's adaptation of proxemics into Human-Computer Interaction introduces technological devices as interactors in a ubicomp ecology, which separates them from fixed and semi-fixed features (2013). Appliances, however, encompass a larger number of devices with different sizes, form factors and functions. Thus, a binary categorization such as fixed versus semi-fixed is overly constrained, as most appliances would be considered *semi-fixed*. Consequently, it is necessary to adopt a more nuanced view of appliance mobility. In particular, appliance mobility can be seen as fitting on a spectrum ranging from portable (highly mobile) to movable (mobile) to static (rarely moved), as shown in Figure 3.2. To explain, the size, weight, purpose and location of an appliance affects how

they can be considered in terms of mobility. A light switch attached to a wall is permanently fixed and thus considered static. However, a 50" television can also be considered static due to its weight and functionality. Most households place televisions at a specific location in the room, typically chosen to optimize comfortable viewing, and leave it at that location unless they are reconfiguring the room. It may even be 'permanently' mounted on a wall. At the other extreme are very light appliances that are *portable*, such as a small Bluetooth speaker. Appliances such as these are often relocated for convenience: one can use it to listen to music in the living room, or even move it to the bathroom while taking a shower. In between are somewhat light appliances that are *movable*, such as a floor lamp. While they tend to be left at particular locations, people can reconfigure them in the room with relative ease. It is worth emphasizing, however, that size and weight are not the only factors to consider when thinking about an appliance's mobility. For instance, a router can be considered a *movable* appliance, yet routers are often placed at a specific location as they are required to be within range of an Ethernet port, and unplugging it could mean that it must be reprogrammed. As a result, the router can only easily be moved within the limited range of the length of its attached cables. Figure 3.2 also shows where in the spectrum the aforementioned appliance examples can be placed.

The different degrees of mobility of different appliances can affect the application of proxemic remote controls. In order to enable people to interact with these appliances, the infrastructure of the system must know the locations of the appliances. For instance, the location of a rarely-moved appliance, such as a television, can be configured in advance, while a highly movable appliance may require location tracking. Discoverability is easier for appliances that are rarely moved, as people quickly learn their location. In contrast, for mobile appliances, discoverability is more difficult as a person must visually search for them – they may have been moved since their last use.

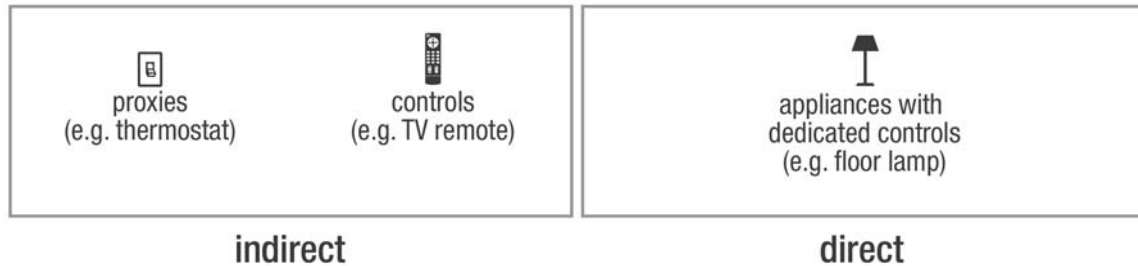


Figure 3.3 Categorizing appliances according to the directness of their interaction.

3.1.2 Directness of Appliance Interaction

The second appliance attribute concerns the directness or indirectness of a user's interaction (Figure 3.3). At one extreme are appliances that afford *direct interaction*, in which an appliance is operated via controls located physically on it, and the current state of the appliance can be ascertained by looking at it. Examples include floor lamps and portable radios, with which the user can directly engage via their physically embedded controls, such as an on/off switch or tuning knob.

At the other extreme are appliances that afford *indirect interaction*. In this case, the appliance serves as a proxy to something else. An example of this is a thermostat, which is a proxy to a larger infrastructure: the central heating and cooling system of the entire house. It is possible to interact with this infrastructure indirectly via the thermostat. The thermostat serves as an indirect appliance: it controls and exhibits information about the heating and cooling system, such as indicating the current temperature setting (which the user can alter), the temperature of the room, and (depending on how sophisticated it is) a schedule regulating temperature over time. Indirect appliances allow people to interact with multiple related objects in a simple manner, such as a bank of light switches to control multiple lights located in the room; or operate appliances that are beyond physical reach, such as a dimmer switch controlling a ceiling fan. In one sense, a traditional remote control can also be considered to be an indirect appliance that controls an appliance at a distance.

3.1.3 An Appliance's Physical Manifestation

While the previous dimension focuses on the interaction that takes place between a user and the appliance, physical manifestation refers to the form factor of the appliance itself. An appliance is considered *visible* when a user can see the entity that performs the function he/she wishes to control, such as a television or a printer. A *proxy* is an intermediate physical device that operates another appliance, such as a light switch or a thermostat. Finally, some appliances do not have a physical manifestation and thus can be considered *virtual*, as they exist only as a digital object. These can often be only accessible through devices such as a PC or mobile device. An example of this is Russound¹, which is a sound system for smart homes that includes multiple speakers throughout the house. The speakers are often located in or near the ceiling and can be difficult to spot. The only way to control audio played through these speakers is via an application only for mobile devices, which connects to a server and amplified that are hidden from view. To interact with these particular appliances, a system design can consider anchoring these virtual controls to a spatial location and thus take advantage of a user's spatial memory (e.g. room entrance).

3.1.4 Appliance Groups

This thesis primarily focuses on how to interact with individual appliances, but it is worth considering that some appliances operate as a group. There are two ways of thinking about appliance groups. First, spatially separated appliances can work collectively to perform the same task and are usually controlled by a proxy. One example of this is lighting systems, which are controlled by flipping a light switch that activates or deactivates multiple lights. Second, appliances can operate together, each performing a different but complementary task. An example of this is a typical home theater system, containing a television, a video player, a game console and an audio system. Although these appliances are disparate, they are

¹ <http://www.russound.com/>

collectively considered as part of an entertainment unit, its components are usually clustered together and physically connected through wires and share resources. From a remote control point of view, these collective appliances can be considered as belonging to a hierarchy, in which control can occur on different levels. Using the home theater example, the theater as a whole can be considered a unit that performs certain universal tasks, such as switching it on or off, adjusting sound volume, and switching between audio and video input sources. The individual components perform more specialized actions, such as changing the channel on the television. This hierarchy structure can also be applied to an ecosystem of devices that coexist in the same space without being part of the same system. For example, a person could enter a room, hold up their tablet, and get a list of devices present in the room with some of their state information.

3.1.5 Appliance Complexity

The last appliance attribute is complexity, as considered by both what functions can be controlled and the number of states it can assume. An appliance can be *simple*, such as a lamp that only has an on/off switch. On the other hand, a television with a cable box has many controls and states, which make this a *complex* appliance. Appliance complexity varies considerably between these two extremes. For instance, a battery-powered alarm clock has multiple controls and/or states: displaying/changing the current time, the alarm time, and the alarm status (on/off), and the state of the battery life.

An appliance's complexity is also affected by whether these states are concrete and thus visible, or abstract. For example, the state of a light is concrete, as one can see whether the light is currently on or off. However, a schedule that controls the light is abstract, and thus must be presented visually, typically on a console with a display on it.

Another aspect that adds to complexity are the kinds of controls required, which in turn depends on the parameters that can be adjusted. For example, a light switch controls a simple binary parameter. In contrast, the volume of a radio is a

continuous parameter, which may require a dimmer control (a sliding bar or a rotating knob). Some controls are highly specialized for the task, such as an interface for setting a schedule, as originally stipulated by Plaisant et al. (1992).

An additional complexity to consider is an appliance that features transferable content. For instance, the appliance may contain an instruction manual or warranty that can be downloaded and transferred to a mobile device. Conversely, information could potentially be transferred from a mobile device to the appliance, for example, sending a file from the mobile device to a printer.

The aforementioned complexities can indeed affect remote control interface design. As shown in Chapter 2 (§2.1), most traditional remote controls usually have many physical buttons that can change the state of a particular appliance without revealing their current state. More complex digital appliances may require better interfaces than banks of buttons and need to provide corresponding visual indicators. Furthermore, an appliance may have many abstract (invisible) states, such as the appliance's energy consumption, and the end user will need some means to examine and adjust those states.

3.2 Operationalizing Proxemic Interaction

The previous chapter (§2.3) introduced the concept of proxemics and how it has been applied within ubicomp systems as *proxemic interaction*. In the context of this thesis, proxemic interaction serves the same purpose as previous work in ubiquitous computing: it is a means to spatially anchor information and/or content to physical objects, and enables people to leverage space for interaction while preserving socially acceptable protocols. Thus, they describe the interaction flow and inform the interaction and interface design. Figure 3.1 on page 42 shows the different components of proxemic interaction which will be further described in this section.

An effect of applying proxemic interaction for remote controls is that the controls become situated onto individual appliances, and the spatial relationships between the mobile device and the appliance dictate the amount of contents revealed. This means that an appliance's specific task that is typically performed through a computer or mobile device can be spatially situated on that appliance's physical component. For example, one could print a file by approaching the printer with a tablet computer and dragging and dropping a file on the printer interface control, as opposed to navigating menus and dialogs on a traditional PC that may be physically separate from the printer itself.

Given that the design constraints for an ecology of appliances have different constraints from those of traditional ubicomp ecologies, which focus on interactive surfaces (e.g. large displays, tablets), these principles need to be further fine-tuned. The focus of remote control interaction is that a user holds a mobile device to interact with a large ecology of appliances. The next subsections will look at how proxemic variables can be leveraged to this end and how to adapt gradual engagement for interacting with an ecology of appliances through a mobile screen. Finally, the last section describes how to contextualize gradual engagement in order to present content through a mobile device.

3.2.1 Leveraging Proxemic Variables

This section is concerned with how proxemic variables are used in the context of appliance interaction. Proxemic variables were originally proposed by Marquardt (2013) and summarized in Figure 3.4 as means to inform the design of proxemic interaction. The variables represent building blocks that can be understood by people and by sensing technologies to make decisions as the result of the relationship between two entities (e.g. person-to-person, person-to-device, device-to-device). An example of this is a large display that reacts to a person's distance from it. These are two entities with different *identities*, a person and a large display. When the person faces the display (*orientation*) and moves closer, the display reveals more information about a bulletin board. In this case, the change in

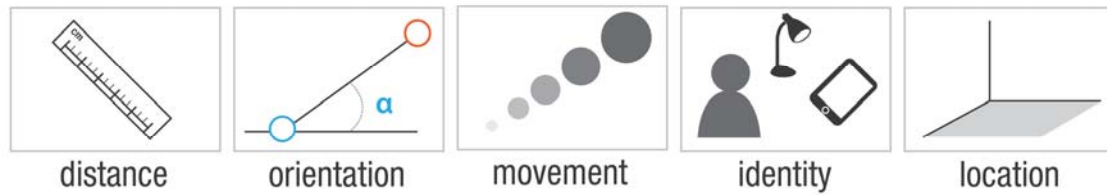


Figure 3.4 Visual summary of proxemic variables

distance between the person and the display makes the system react accordingly. This subsection describes the different variables as defined by Marquardt (2014) and how they are leveraged in this particular context (interaction with appliances via a mobile device).

Distance is used as a measure for determining the level of engagement between the individual and the appliance they wish to interact with. Distance is often directly mapped to engagement. This mapping can be *discrete*, meaning that different zones or distant thresholds trigger different stages of interaction, revealing more content and interaction opportunities on the mobile device; or *continuous*, in which content is revealed progressively on the mobile device as a function of distance.

Orientation refers to the direction that an entity is facing with respect to another. It serves as a mechanism to determine if (1) the person is engaging with a particular appliance, and (2) which appliance is the current center of attention. This way, the system can discriminate between which control interface should be presented on the device.

Movement is the change of position or orientation over time. Considering movement, one can understand aspects such as velocity and acceleration. In this thesis, movement is a design variable that is used implicitly, and appear in two scenarios: (1) a person approaching an appliance changes their level of engagement as a function of distance, which means that changing the speed at which the distance increases or decreases (i.e. walking faster or slower) will change the speed at which the interface is presented on the mobile device; and (2) changes in orientation will change which objects are presented on the mobile device's screen.

Movement incorporates an understanding of the directionality of the engagement, meaning that the system can understand if a person is moving towards an appliance (engaging) or moving away from it (disengaging).

Identity is a way to uniquely describe the different entities in the space. In this context, identity is constrained to three types: people, mobile devices and appliances. The identity of the person can influence the types of control and the information presented. Discerning between individuals can enable customized controls, such as specific people having more specialized controls if they own the room, or parental controls. Mobile devices are tracked continuously and understand their relationship with other appliances in order to determine what appliance the user is interacting with. Appliances are the devices the user wishes to interact with. Each appliance contains a different interface and information that is presented on the user's mobile device, in which they are instances of a particular device and differentiate themselves depending on their complexity and customization.

Location reflects the qualitative aspects of the space that define the rules of social context and behavior. Location affects the previous variables as a mechanism to establish the rules for interaction. For example, the physical constraints of the space can affect the relative measure of proxemic distances. This contextual information may also influence the role of identity, such as determining which appliances can be grouped or are working together as a function of their spatial relationships, or who the owner of the room is, as well as which persons have different control constraints.

3.2.2 Gradual Engagement of Controls

As mentioned in the previous chapter (§2.3), Gradual Engagement is a design pattern that describes how engagement increases as a function of proximity. That is, more digital content is displayed on a user's mobile device as he or she moves closer to the device he or she is interacting with. Based on this, information is presented in three stages: awareness, progressive reveal and information transfer.

As mentioned in the previous subsection, there are two ways in which engagement take place: first is through discrete stages (each distance threshold triggers a different level of information as a layered approach), and the other through continuous (information gradually appears or disappears with animated transitions as a function of distance and not as a function of time). This work focuses on continuous engagement since the animations are a function of distance, however, these interfaces are broken down into transitioning steps. This can be seen in Figure 3.5.

The next subsections take a deeper look into gradual engagement and how it can be used to control appliances. This is illustrated through the types of engagement that can occur, the information flow mapped as a function of distance, and the rationale to present the information on a mobile device.

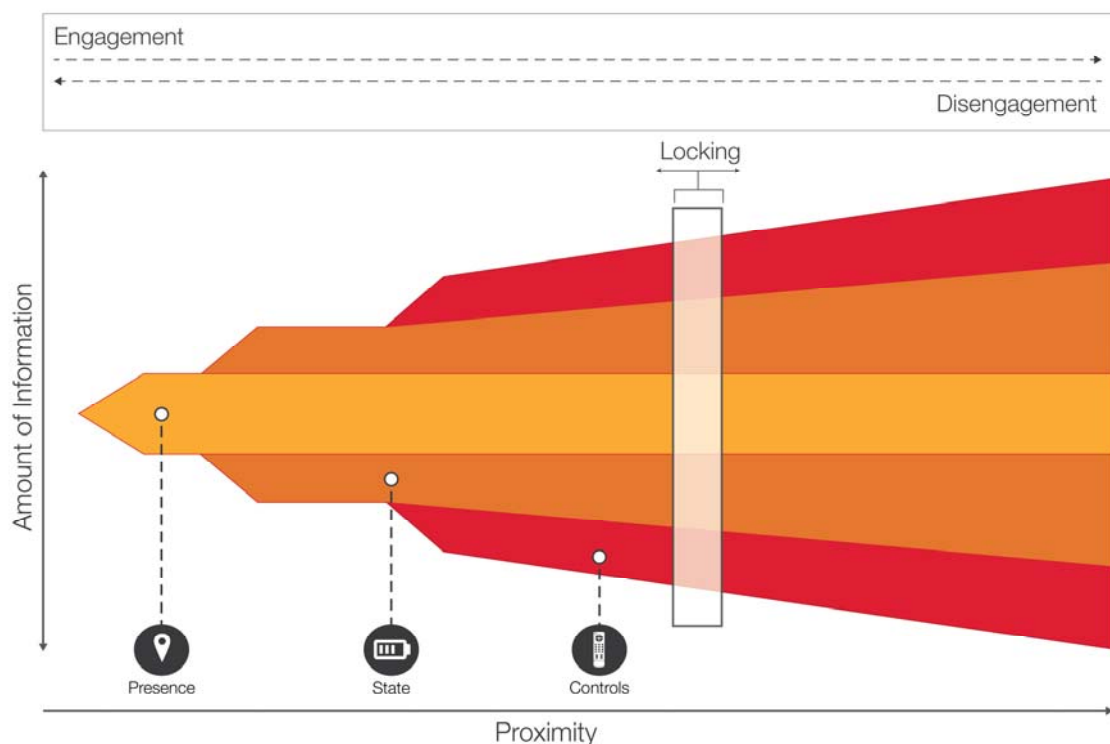


Figure 3.5. Gradual Engagement of Controls and its content flow.

3.2.2.1. Types of Engagement

There are three main forms of engagement in terms of interacting with appliances: (1) engaging, (2) disengaging, and (3) manual override or locking. While gradual engagement originally explored the amount of content revealed as a function of distance, these three types of engagement that occur can help in understanding in depth how the process takes place.

Engagement. Engagement occurs when a person faces and moves toward a target, that is, they are gradually engaging with it. As the person approaches the target they wish to interact with, they likely wish to see more content related to that target on their mobile device, which can take the form of information or controls depending on the appliance complexity and interface design.

Disengagement. Disengagement takes place when a person moves away from a target or appliance. Two cases are possible: The first is when the person is moving away from the target while still oriented towards it. This causes the system to apply a reverse function of the engagement, meaning the information becomes less complex (i.e. fewer controls and less information about that appliance are displayed) until it disappears. The second case occurs when an individual already is in a particular level of engagement with the target and decides to face away from it or reorient towards another target. This can be interpreted as the user shifting their focus or interest. Instead of gradually hiding the information, the mobile device transitions towards the new focus of attention at the current engagement level as dictated by the distance relation.

Manual Override or Locking. Constraining interactions to the gradual engagement pattern can be restricting, as this requires the user to always physically face and approach an appliance in order to interact with it. A shift of focus may happen accidentally if the user's center of attention changes due to small movements on a mobile device. These can occur when the user moves their mobile device while getting close to a thermostat and reorients the device to hold it more comfortably or accommodates for reflections. Gradual engagement can also be

restricting for users who wish to remain stationary (i.e. sit down) and still be able to control an appliance. Manual override, or locking, provides a way to relax this principle and perform ad-hoc modifications of any appliance in the room. This means that users are able to (1) pause the current spatial interactions, (2) manually change the level of engagement, and (3) select any appliance from the ecology and engage with it. This relaxation also enables transitioning to other techniques discussed in Chapter 2 §2.3.1: scanning through manual selection of an individual appliance from an overview (described in next section); touching by approaching a digital appliance to retrieve content; and pointing by focusing on an individual appliance through device orientation and manually locking and adjusting the level of engagement.

3.2.2.2. Types of Content

A critical aspect of remote control design is to give people opportunities to see and make sense of appliances. The digital content placed on an appliance can be organized into three categories: *presence*, *state* and *controls*; and each has implications for remote control design. Figure 3.6 illustrates these components along with some examples of each. It is worth noting that this categorization is not necessarily clear-cut, but it provides a basic dependence which will be explored more in depth in the next subsection.

Presence. Presence information refers to the basic identifying information of an appliance (Figure 3.6, left). At a high level, an appliance can be thought of as having a name and a location, but this can be further extended by more fine-grained descriptions, such as a globally unique identifier (*GUID*): a visual icon that represents the appliance, manufacturer, and even type of appliance.

State. State refers to information that describes the current condition of the appliance (Figure 3.6, middle). This can be the result of previous actions and controls, or simply the result of current sensor readings, such as a thermostat showing the current temperature of the room. Some state information is immutable, meaning it cannot be changed through remote controls (e.g., battery

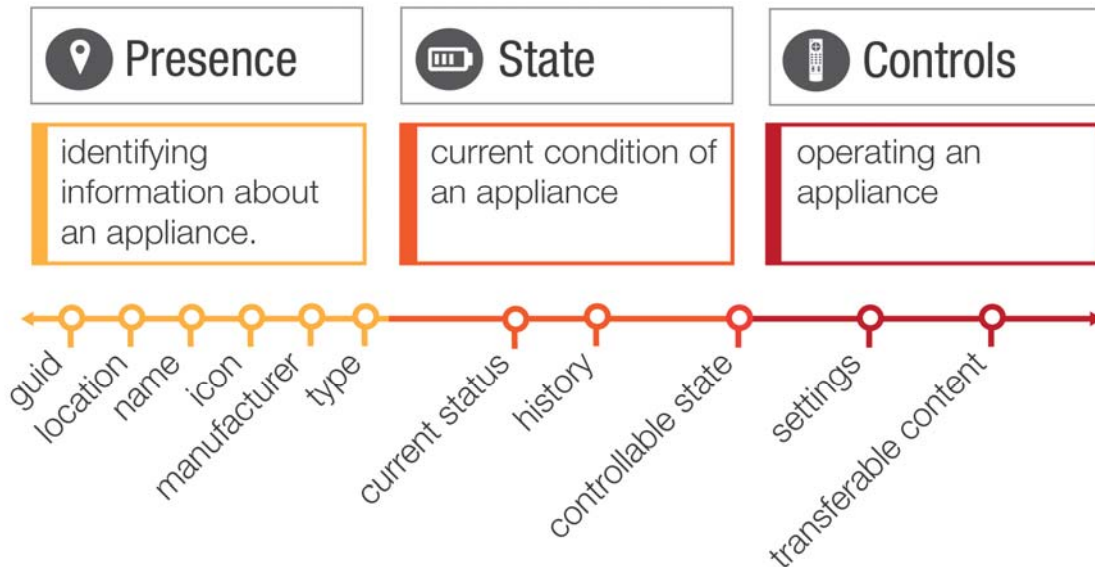


Figure 3.6 Diagram the different types of content, which encompasses three categories: presence, state and controls – these become progressively more complex and their purpose becomes more specific.

levels). State information can go beyond showing the current state, such as revealing energy consumption over time, or displaying a history of actions performed on the appliance. A remote control needs to be capable of displaying such states to provide awareness to the end user.

Controls. Some appliance states are controllable (Figure 3.6, right). The most basic definition of a control is the capability of changing the current state of an appliance. These controls have varying levels of complexity depending on the functionality: a very simple control switches an appliance on or off, while more fine-grained controls allow for discrete values such as a light dimmer. More complex controls enable higher customization through settings (e.g. scheduling). Some of these settings can be saved, such as favorite channels on the television. Other controls may require information transfer between the mobile device and the appliance (e.g. printing a file).

3.2.2.3. Content and Interaction Flow

Another important aspect when considering gradual engagement is the type of content that is being displayed on the mobile device. This work is not intended to

prescribe a definite process for design of gradual engagement, but there is a flow or ordering that must be taken into consideration. It is important to keep the different types of content in mind, shown in the previous subsection: presence, state and controls. There cannot be state information if the system has no knowledge of the device that the user interacting with (*presence*). Showing *state* information can facilitate *controls* as users can transition from seeing a state to being able to modify it. As a result, a typical transition in gradual engagement would be to go from presence information, then to state and finally to controls. The information should build up and increase in complexity as the user approaches an appliance. By making each interface item build up over time, one can ensure a smooth transition from a simple interface to a more intricate and flexible one. This can be demonstrated in the case of a floor lamp in the ecosystem. From afar the user can see a representation of a lamp on their mobile device that immediately shows interaction possibilities – the opportunity to turn it on or off. The current state of the lamp (off or on) must be indicated to the user in order to be able to effectively communicate possible actions. Closer proximity between the lamp and the user can reveal finer controls on the mobile device such as a dimmer switch. This example will be revisited in the implementation section, see §4.2. Here, while there is a dependence structure to how the information appears (e.g., there can be no controls without state information, and no state information without the presence of the device), a specific design can bring forth simple and relevant content forward to the main user and leverage presence, state and controls as starting points for content flow. Figure 3.5 on page 52 shows a general depiction of how the interaction flow takes place for one appliance in the ecosystem.

3.2.3 Presenting Information on a Mobile Device

Unlike traditional user interfaces, proxemics take *spatiality* into consideration. This means that user interfaces have the opportunity to be dynamic and mutable as one moves around space. This subsection describes some traditional user

interface constructs that can be used to inform the design of remote controls for a ubiquitous computing ecology.

Ben Shneiderman introduced what is known as the information visualization mantra: “Overview first, zoom and filter, then details on demand” (1996). This concept is highly influential in the fields of information visualization and user interface design as it establishes building blocks for revealing content and preserving context. This rationale can be transferred to interaction with an ecology of appliances in the same way, which addresses the original problem statement and goals from Chapter 1. People have to be able to *discover* interactive devices (overview), be able to *select* one among the ecology (filter), and then *view information and controls* (zoom and details on demand).

3.2.3.1. Overview

In this context, overview corresponds to providing people with a sense of what interactive appliances are present and their relative positions. Given the fact that this thesis uses spatial interactions (proxemics), the design of these controls should empower people with means to be aware of nearby devices any time. The way this can be achieved on a remote control on a mobile device is by providing visualizations that are readily available at a glance.

In principle, an overview is all of the basic information displayed within an interface or visual representation of data while being zoomed out (Shneiderman, 1996). For example, having a small overview as a secondary representation (e.g. on the edges of the screen) allows a person to see relevant content while still preserving a general sense of other interactive items, in this case, the different appliances in the ecology. As a result, presenting spatial references provides a way to enable that discoverability. Previous work in ubiquitous computing has mostly presented spatial references with relative positioning as a bird’s-eye view (Gellersen et al., 2009; Marquardt, Ballendat, Boring, Greenberg, & Hinckley, 2012), which can be seen as an influence of this thesis work when the user’s mobile device is positioned parallel to the ground. Building on this, some work in

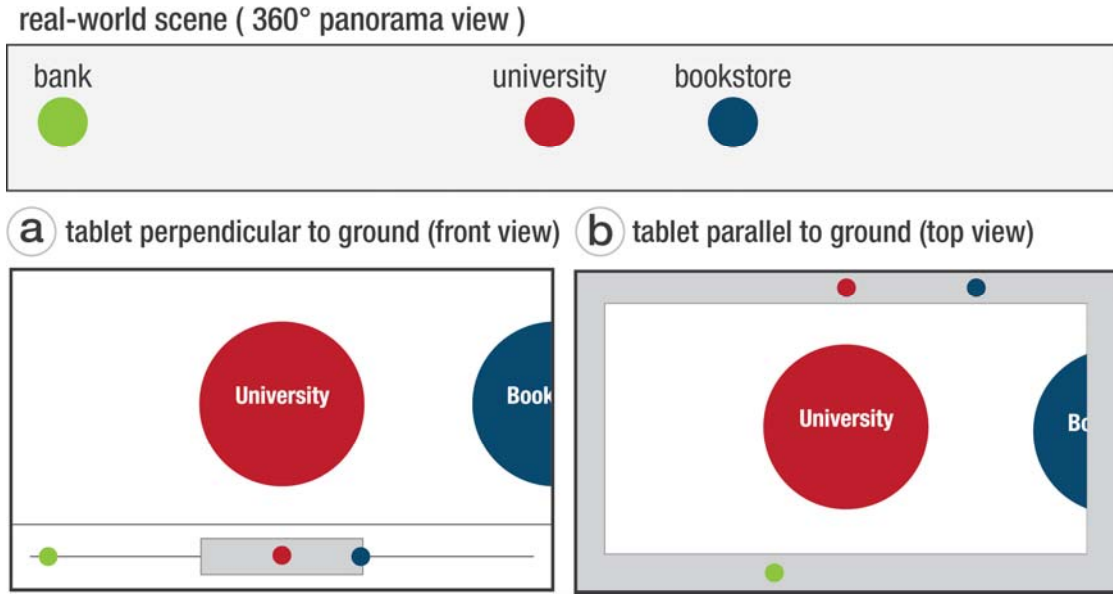


Figure 3.7 Visual representation of spatial references showing: (a) a panoramic overview as proposed by Leihokinen et al. (2002); and (b) top-view spatial references, based on Gellersen et al. (2009).

augmented reality has examined ways to represent other physical objects in space that are not in view as means to provide context. One of these techniques was proposed by Leihokinen et al. (2002), in which they assume the mobile device is perpendicular to the ground so that the camera feed can reveal the information about the real world with digitally enhanced annotations, such as the name of the place. In this case, they use linear mapping to represent off-screen targets: the closer they are to the center, the more they are aligned with the center of the user's field of view. These two types of spatial references are ones used in the design of this thesis work and are illustrated in Figure 3.7.

While these two specific techniques are implemented in this work, and can be switched between depending on the tablet orientation, this adoption of techniques is not meant to be the only solution. There are other means of providing overview – such as using maps with absolute positioning, or shrinking the scene to reveal all the information (Mulloni, Dünser, & Schmalstieg, 2010). However, our chosen techniques have been selected on the basis that they (1) separate the overview from

the content into two different modules and (2) provide the overview at the same time as the content.

3.2.3.2. Filter

As mentioned previously in this chapter (§3.1.3), the relationship of the orientation between the mobile device and the appliances can determine which appliance one intends to interact with. As a result, orientation can be used as a means to filter: the interface can display the appliances in the user's field of view that they can interact with. This means that the focus is on the objects that the user is facing. When the user changes their orientation, the position and availability of appliances to interact with will change relative to this. In this specific design, one appliance is revealed at a time in order to allow individuals to perform touching and sliding interactions with each control. Some other augmented reality approaches display multiple objects at once, such as He's (2010) visualization of energy consumption, in which the energy consumption for each object was overlaid on top of a camera image.

3.2.3.3. Zoom and Details on Demand

The distance or proximity between the person and the appliance is a metric that can be used as a mechanism to reveal more content, or zoom. Zoom need not be restricted to sizing – it can also serve as a semantic zoom (Bederson & Hollan, 1994): the amount of content available to the user, in this case information and controls, increases as the distance decreases. This idea is known as 'Zoomable User Interfaces' (ZUIs). Previous work in proxemic interaction applied this idea by showing more content on a display as people got closer, or by enabling content transfer between nearby devices (Ballendat, Marquardt, & Greenberg, 2010b). Similarly, in this thesis, as one approaches a particular appliance, the interface changes dynamically and provides more detailed content. This will allow content flow from simple to increasingly complex, and it can be a mechanism to relax the flexibility versus usability trade-off (Lidwell, Holden, & Butler, 2003). This trade-off describes that by increasing the flexibility of the interface, it automatically

introduces more complexities that make it less usable. While gradual engagement can bridge between simplified usable interfaces and highly flexible controls, it can still be difficult to present a large array of controls in close proximity to an appliance because of the size of a mobile device's screen. In order to address this complexity, we can apply *micro-mobility* (Marquardt, Hinckley, & Greenberg, 2012), meaning that some of the controls could be potentially distributed in the space around the appliance either through spatial references (above, below to the sides of the appliance) or utilizing specific parts for an appliance as a way to reduce screen navigations and menus, such as approaching a radio's speakers to get volume controls. This schema can also apply to interacting with groups of appliances: the user can apply controls for the whole group and then branch into individual appliances, as mentioned in §3.1.4. The intention is to reduce the number of manual navigation and paging required to find a specific option.

Going back to Figure 3.1 in page 42, the conceptual framework for proxemic-aware controls presented in this chapter serves to structure the variety of appliances that can be controlled. It also explains how proxemics interaction can be applied for the design of remote controls in a ubicomp ecology. Gradual engagement of controls frames the interaction flow between a person and an appliance, in which a mobile device acts as the interface between the two. Finally, our application of presentation techniques from traditional user interfaces operationalize how gradual engagement takes place within the mobile device. The next chapter takes these concepts and shows them implemented as prototype that demonstrates how this framework can be applied.

Chapter 4. Proxemic-Aware Controls in Action

“In analyzing and designing systems and software we need better means to talk about how they may transform and/or be constrained by the contexts of user activity: this is the only way we can hope to attain control over the ‘materials’ of design.”

- John Carroll in “Five Reasons for Scenario-Based Design”

Based on the considerations presented in the previous chapter, we designed and implemented a suite of *proxemic-aware controls* as they might appear in a domestic environment containing various digitally-aware appliances. Our system is a proof-of-concept. Its primary purpose is to illustrate how different interaction explorations enable us to **use socially established protocols (proxemics) as a means to seamlessly reveal information and controls for appliances in a room**. This is done by first outlining the different appliance interfaces (§4.1), and then illustrating various concepts through usage scenarios (§4.2, §4.3). Technical and implementation details of the underlying system will be presented in Chapter 5, which will explain the prototyping environment, nuances of the visual representations used, and the system architecture, as well as its limitations.

4.1 The Appliance Interfaces

The implementation seeks to illustrate the different design variations within appliances following the framework in §3.1. Consequently, we prototyped a home

Chapter 4 • Proxemic-Aware Controls in Action

environment with different controllable appliances, each varying some aspect of: mobility (§3.1.1), directness (§3.1.2), physical manifestations (§3.1.3), grouping (§3.1.4) and complexity (§3.1.5). This section briefly describes features of these appliances and their corresponding interfaces. Figure 4.1 provides an overview, and it depicts our design decisions for each of the appliances and how they are described according to the aforementioned attributes. This set of variations will be revisited more in depth in §4.1.7. The remainder of this section shows and explains







| |  thermostat |  lamp |  radio |  router |  printer |  room viewer |
|-------------------------------|--|--|--|--|---|---|
| MOBILITY | | | | | | |
| static | | | | | | |
| movable | | | | | | |
| portable | | | | | | |
| DIRECTNESS | | | | | | |
| direct | | | | | | |
| indirect | | | | | | |
| PHYSICAL MANIFESTATION | | | | | | |
| visible | | | | | | |
| proxy | | | | | | |
| virtual | | | | | | |
| INDIVIDUAL VS GROUP | | | | | | |
| individual | | | | | | |
| group | | | | | | |
| COMPLEXITY | | | | | | |
| low | | | | | | |
| medium | | | | | | |
| high | | | | | | |

Figure 4.1 Appliance Rationale – this figure shows how each of the appliances fall into different categories based on the appliance attributes described in Chapter 3.

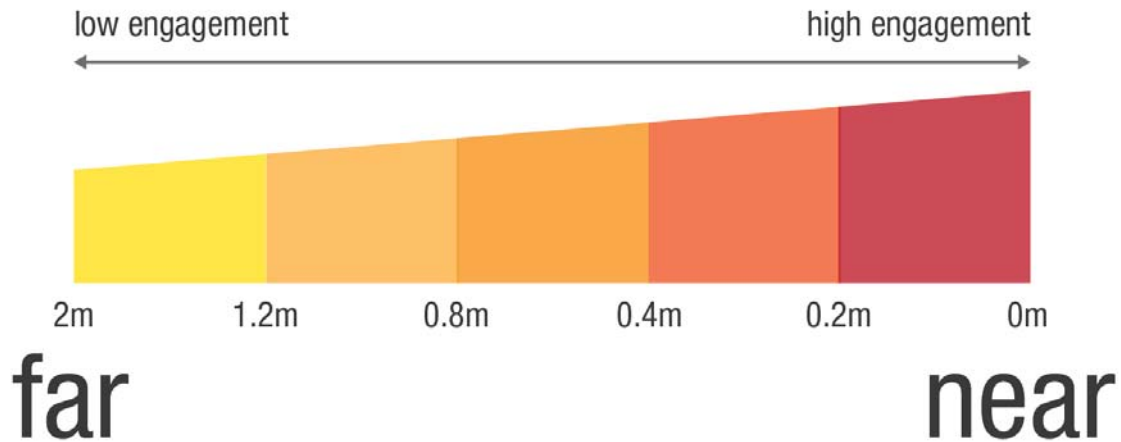


Figure 4.2 General legend for different appliance interfaces. This legend will be used in the next set of figures

the interfaces for each appliances, so that the scenarios of use showcased in the subsequent sections (§4.2 and §4.3) can be more easily understood.

For each appliance below, we introduce different stages of the interface design as they change as a function of proximity. Each interface appears on the user's handheld device (e.g. tablet) when it is oriented towards the appliance. However, the particular interface will adapt based on how far the device is from the appliance (see next section). Figure 4.2 shows a general legend for upcoming figures as we illustrate each appliance's interface at five approximate distances ranging from 2 to 0 meters; some appliance interfaces may omit one or more of these thresholds. Following gradual engagement, it follows that the least amount of content is visible when the user is far from the appliance, with increasing amounts of content and interaction opportunities revealed as one approaches the appliance. While the subsequent figures show specific states, it is important to note that the interface responds continuously, in which the content and the size of application controls change as a function of distance.

4.1.1 Viewing and Interacting with Appliances

Before delving into the individual appliances, it is important to describe how the general interface operates. As shown in Figure 4.3, people control the system using

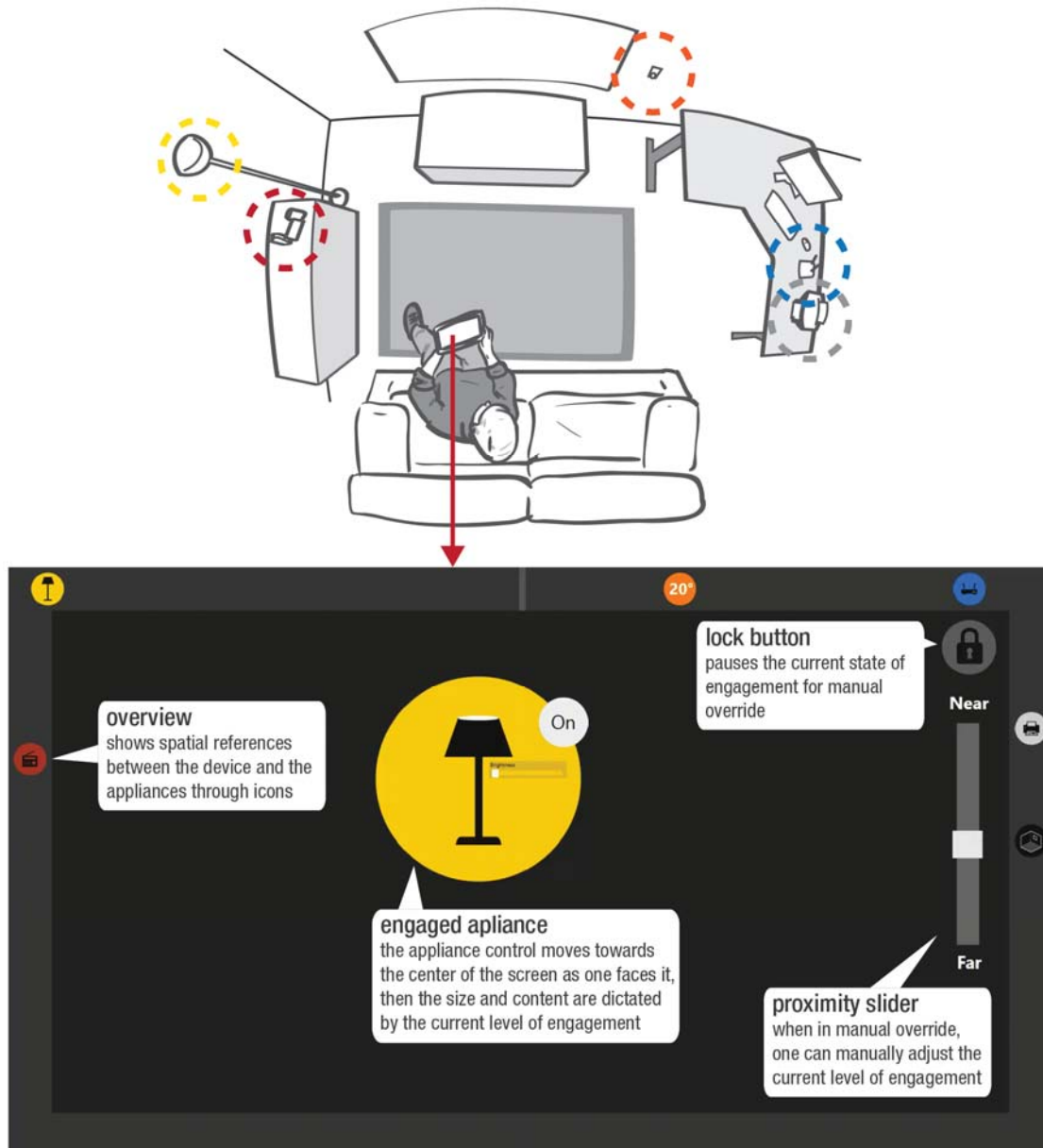


Figure 4.3 Our system interface in action. The user holds the device and moves around the room to navigate through different controls, thus updating the interface dynamically.

their mobile device by spatially navigating around the room and viewing different controls that update dynamically according to their proxemic relationships (e.g. distance, orientation). A large portion of the screen is used to reveal controls of appliances, which change in size and amount of content as the user physically approaches them. On the top-right corner, one can see a lock button that enables manual override, further elaborated in §4.2.3. If the interface is currently locked,

a slider is revealed, which allows individuals to change the current level of engagement of an appliance. On the edges of the screen there is a room overview depicting the current spatial relationships to all interactive appliances in the room through icons. More specifically, our two types of overviews will be further described in §4.2.2 and also relate to our design rationale in §3.2.3.1.

4.1.2 The Thermostat

When the thermostat interface is first revealed, it initially shows only the current temperature of the room (Figure 4.4 step 1). As the device (tablet or phone) moves closer (1.2 m away), the thermostat interface changes to include a gray semi-circle that indicates the current temperature setting that the room is adjusting towards (Figure 4.4 step 2). As the device moves even closer (0.8 m away), the temperature setting changes to blue to indicate that it can now be controlled interactively (Figure 4.4 step 3, top). A user can now drag the blue circle to change the room temperature setting, which is temporarily reflected on the interface to accommodate for finger occlusion (as illustrated in Figure 4.4, step 3, bottom).

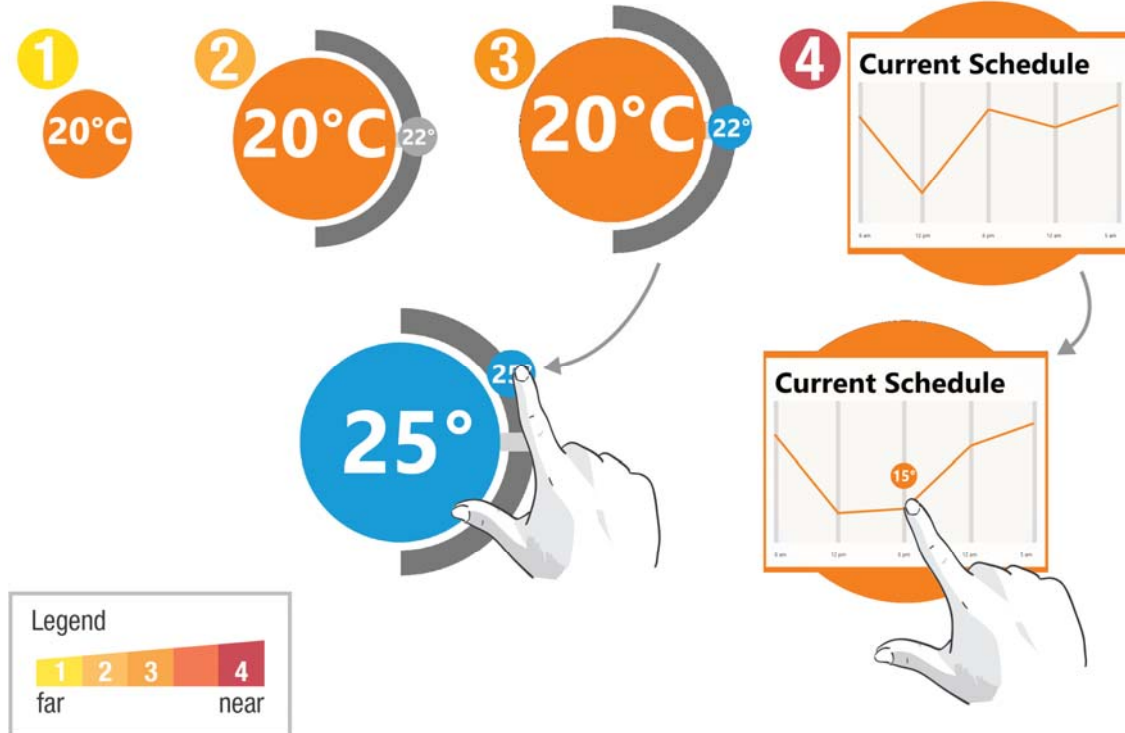


Figure 4.4 Thermostat interface and its progression as a function of distance.

When the person moves very close to the thermostat (0.2 m away, shown in Figure 4.4 step 4), an interactive schedule appears that displays set points for the temperature over the day. This schedule can be manually modified through a dragging gesture.

4.1.3 The Lamp

The second appliance is a lamp that lets users select its on/off state, control its brightness, monitor its energy consumption, and activate its power-saving function. The on-screen lamp interface uses two different colors to reflect the current state of the lamp: gray for 'on' and yellow for 'off'. When the tablet detects that it is far from the lamp (2 m), the lamp interface displays the lamp's current state along with an on/off button (Figure 4.5 step 1). As the tablet moves closer to the lamp (1.2 m), the lamp interface reveals a slider that offers fine-grained control of the lamp's current brightness setting (Figure 4.5 step 2). As the tablet approaches the lamp further (0.8 m), the lamp interface displays an energy consumption visualization showing the lamp's energy usage over time (Figure 4.5

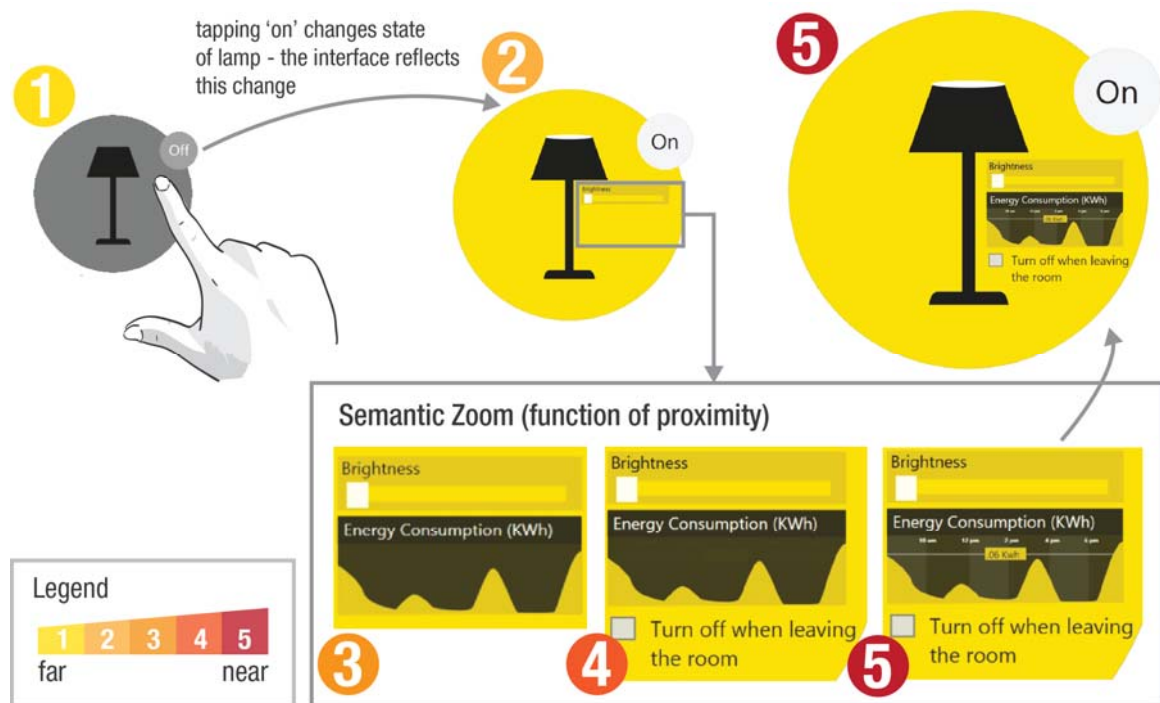


Figure 4.5 Lamp interface and its progression as a function of distance.

step 3). It also includes an advanced power-saving setting that cues the lamp to turn itself off when the system detects that the main user has left the room (0.4 m, shown in Figure 4.5 step 4). When the interface is at its closest proximity (0.2 m), the power usage visualization shows even more detailed axis information: the corresponding times for the consumption and the amounts of Kwh consumed (Figure 4.5 step 5).

4.1.4 The Radio

As the tablet senses that it is approaching the radio, the radio control becomes larger and eventually shows the current time. However, the radio control makes use of adjacent space to provide different controls as means to reduce screen navigations (e.g. menu selection and navigation). This means that the radio shows

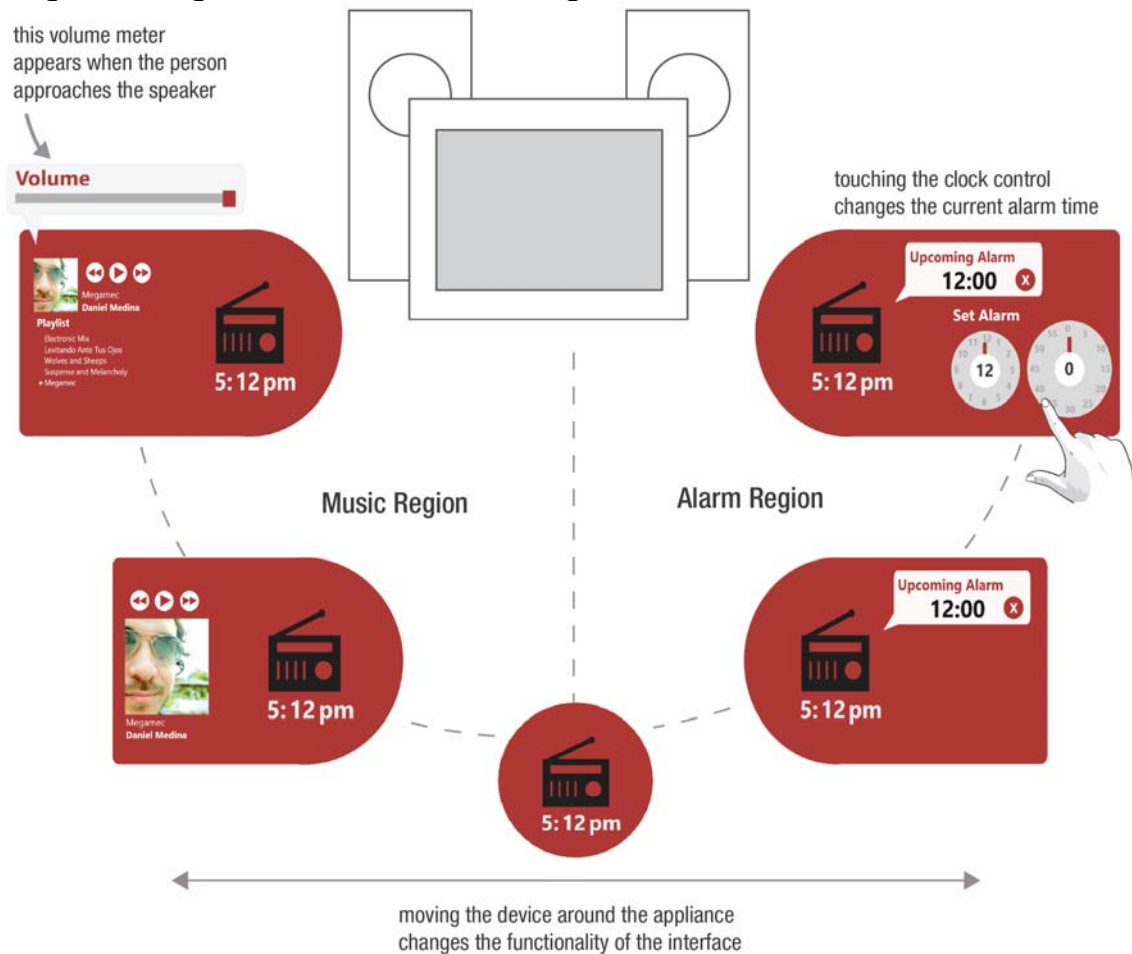


Figure 4.6 Radio interface.

different controls that leverage micro-mobility (i.e. relative positioning) to reveal different content depending on where the user places their mobile device. As shown in Figure 4.6, when the tablet is in front of the radio, the control shows only a circular interface with the current time. However, reorienting the tablet to either side of the radio triggers the circle to expand into additional controls (alarm settings or music), depending on the side. Moving the tablet to the right side brings up the next upcoming alarm, which can be cancelled, as well as an alarm clock control that can be set through touch input. Moving to the left side introduces music controls for controlling music playback and viewing playlist information. Physically approaching the speakers brings up volume controls for the music.

4.1.5 The Router

From afar, the router control simply shows an icon of its presence (Figure 4.7 step 1); as the tablet comes closer, the control displays current upload and download



Figure 4.7 Router interface and its progression as a function of distance.

speeds (Figure 4.7 step 2). As the tablet continues to approach the router, the control reveals connection information about the various devices that are connected to the router, as well as their relative locations (Figure 4.7 step 3). Furthermore, the thickness of the line connecting the router to the corresponding device is proportional to the bandwidth allocated to each device. At its closest distance, the router control displays recognizable icons corresponding to each connected device (Figure 4.7 step 4), as well as several buttons that control advanced router settings (the 'Settings' button) or reset the router back to its default settings (the 'Reset' button). For example, the 'Settings' button opens the router's web interface where settings can be viewed and altered (Figure 4.7 step 4 bottom right).

4.1.6 The Printer

The printer interface shows three main states as a function of proximity. From afar, the printer interface simply displays the printer's presence via an icon (Figure 4.8 step 1). After moving closer, the printer interface adds additional controls to print files from the tablet (Figure 4.8 step 2). After tapping the 'Print File' area, the

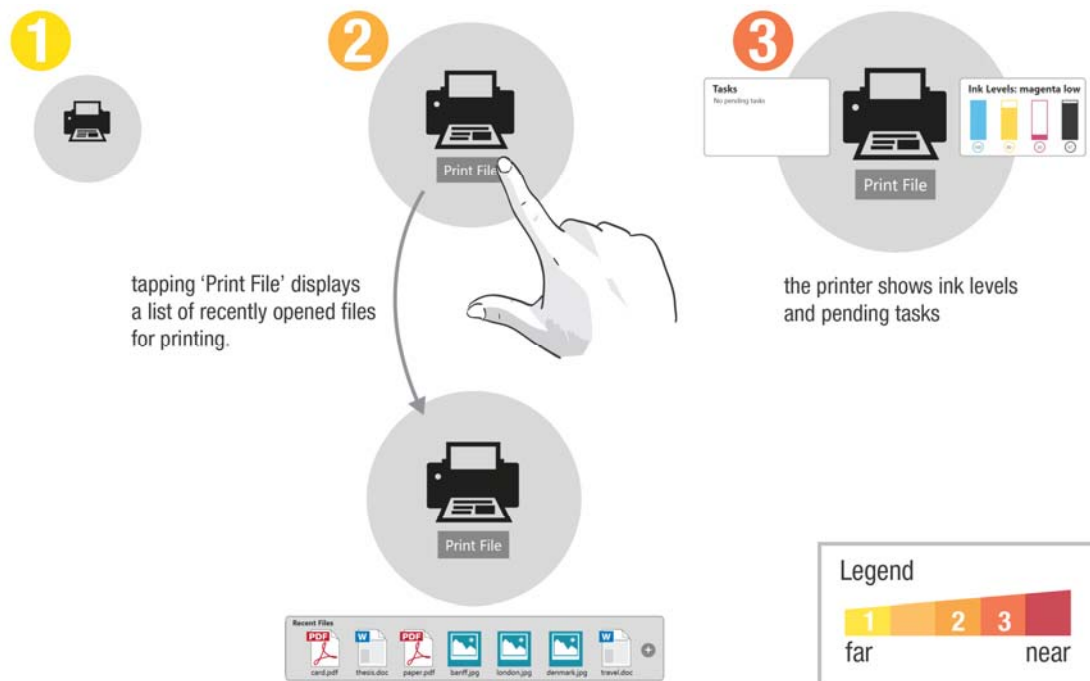


Figure 4.8 Printer interface and its progression as a function of distance.



Figure 4.9 RoomViewer interface showing all of the appliances in the living room.

interface brings up a list of recently opened documents that can be selected for printing. Finally, at close proximity the printer interface shows the list of current print jobs, as well as current ink levels (Figure 4.8 step 3). As we will see in §4.4.3, the printer is also capable of sending notifications, such as announcing when it has run out of ink.

4.1.7 The RoomViewer

The RoomViewer is a control virtually situated at the entrance of the room. It summarizes the controls and states for all interactive appliances in the room ecology as a hierarchy, as one can continue walking past the entrance of the room to engage with each individual appliance. As shown in Figure 4.9, the RoomViewer reveals all of the connected appliances and further provides quick access to some basic controls for several appliances, such to turn the lamps and/or the television on or off. This list of appliances also reveals some basic state information, such as the current room temperature for the thermostat, or the current television channel.

4.1.8 Contrasting the Appliances

Revisiting the different criteria that describe appliances discussed in the previous chapter (§3.1), and referring back to our Figure 4.1 on page 62, we can see that each of these appliances exhibit different features (marked in green). They all have different degrees of mobility, the thermostat and the RoomViewer acting as static, unmovable entities, and having the lamp, the router and the printer as relatively movable appliances, while the radio acts as a portable appliances that is lightweight and can be placed anywhere. The thermostat and the router are connected to larger infrastructures (e.g. central heating) and act as physical proxies that enable controls. Similarly, the RoomViewer is a virtual proxy to the different controls of the room. These proxies thus leverage indirect interaction. However, the interaction with the router can also pertain directly to the appliance itself, such as resetting it, and thus enabling some direct interaction as well. The other appliances: the lamp, the printer and the radio all afford direct interaction. All appliances have a visible physical manifestation, in which the thermostat and the router act as physical proxies, with the exception of the RoomViewer being a virtual proxy. All of the appliances are individual entities. The RoomViewer, on the other hand, acts as a hierarchy that encompasses all of the devices in the room. Finally, as we have seen in the previous subsections, all of these appliances have different degrees of complexity in terms of their functionality, their controls and their transitioning from presence to state to controls –the thermostat and the printer are the most simple, and the radio is the most complex.

4.2 Main Usage Scenarios

We use scenarios to illustrate the application of our concepts. The purpose of these scenarios is to tell a story that conveys the different concepts we are trying to highlight. Each scenario focuses on a particular task, which we use as the title. We then describe the story based on a persona, in this case a young man named Trevor who is in his mid-twenties and lives in a house that has been augmented with

proxemic-aware controls. After each scenario, we show each of the concepts described and how they correspond to the design consideration from Chapter 3. We then discuss each scenario to explain the different nuances of the system along with any potential limitations. Each scenario is accompanied by a storyboard figure that summarizes the different user actions. To better view these concepts in action, please refer to our video figure attached, as well as the previous figures that show details of how the interface of a particular appliance appears at a given distance. Some concepts, such as proxemic variables and gradual engagement are present throughout the different scenarios, though specific nuances may come into our discussion depending on each case.

4.2.1 Discovering Interactive Appliances

Scenario. Trevor walks into his living room and brings his tablet to an upright position (Figure 4.10-1). He rotates his tablet around the room to face each appliance and view its corresponding interface: from the radio on the shelf (Figure 4.10-2), to the thermostat mounted on a wall (Figure 4.10-3), to the router underneath the desk (Figure 4.10-4). In all cases, the tablet shows icons representing each of his appliances that are integrated into the system. When Trevor holds the tablet upright, these icons will appear at the screen's bottom (Figure 4.10-4). If he reorients his tablet horizontally, parallel to the floor (Figure 4.10-5), the visualization changes to an alternate representation, drawing icons around the display's border to indicate which direction each appliance is in from a bird's-eye point of view (Figure 4.10-6, also visible in Figure 4.3). In both representations, as he moves around the room, the position of the icons move accordingly, as each reflects their relative position to the appliance it represents.

Discussion. This scenario realizes several of the concepts described in Chapter 3: overview (§3.2.3.1), orientation (§3.2.1), and filter (§3.2.3.2). More generally, this scenario illustrates how the interface applies these concepts to allow a person to spatially navigate a room to see *what* appliances comprise the room's ubicomp ecology (overview), *where* they are with respect to the appliances (orientation),



Figure 4.10 Main usage scenario 1 - Discovery.

and choose *which* appliance they want to interact with (filter). All this occurs in real time, where information is updated as a function of the person's proxemic relationship (orientation and distance) between their tablet and the surrounding appliances.

There are two ways that the user can get an overview of which appliances are present. First, the smaller icons around border or the bottom of the horizontal tablet (depending on how the tablet is held) provide an 'at-a-glance' overview of all appliances in the room (refer back to Figure 4.3 on page 64 to see enlarged version). Second, the person can scan the room simply by holding the tablet vertically and spinning around in a circle—larger appliance icons (such as those

Chapter 4 · Proxemic-Aware Controls in Action

illustrated in Figure 4.4 to Figure 4.8) animate across the display as those appliances come into the user's directional field of view.

One can infer an appliance's location through orientation information. In the interface, icons serve as a spatial reference, allowing people to bridge the digital information they see with the physical objects in the space. This is done in two ways: (1) to display the small border icons in their spatially correct locations; and (2) to display the larger appliance icon representing the appliance in front of the tablet. In the latter case, as Trevor changes the tablet orientation to face the different appliances, the display reveals the appliance icon using a mix of transparency and animation: the icon becomes more opaque and moves towards the center of the display as the tablet rotates to face the appliance. This will be elaborated upon in §5.2.2.

Finally, this filtering based on tablet orientation allows a Trevor to choose which appliance they want to interact with (§3.2.3.2). While simple, this affords several different strategies: discovering an appliance through the interface, or discovering a physical appliance and then using the interface to view its information and controls. Consider the thermostat as an example. First, Trevor may discover the thermostat by seeing its icon appear on the left side of the display as he scans the environment by moving the tablet from side to side, then look up to see the physical thermostat. Alternately, Trevor may see the physical thermostat first, and then see details of its settings by orienting the tablet towards it. In either case, he can both move his tablet and look to the physical world to confirm that his display corresponds to the thermostat or other appliances in the room.

As a side note, the overview of small icons changes depends on whether the device is oriented vertically (upright) to horizontally. In essence, changing the angle of the tablet alters its spatial referencing. The first horizontal representation is a bird's-eye view that shows appliances around the display's perimeter. This view provides relative positioning of all appliances, highlighting which objects are ahead, to the side or behind Trevor. However, when the device is held vertically,

these relationships no longer hold (e.g., an object at the bottom edge may be interpreted as being on the floor rather than behind Trevor). Therefore, the representation changes to illustrate a linear fish-eye view, with objects in front at the center. These ideas are a direct application of §3.2.3.1.

4.2.2 Gradually Engaging with an Appliance

Scenario. It is late in the day and Trevor feels that his room is feeling too cold for his liking, so he approaches his thermostat to reset the temperature (see Figure 4.4 on page 65 for detailed views of the controls shown in Figure 4.11). He walks towards the thermostat from the entrance of the room and first sees the current

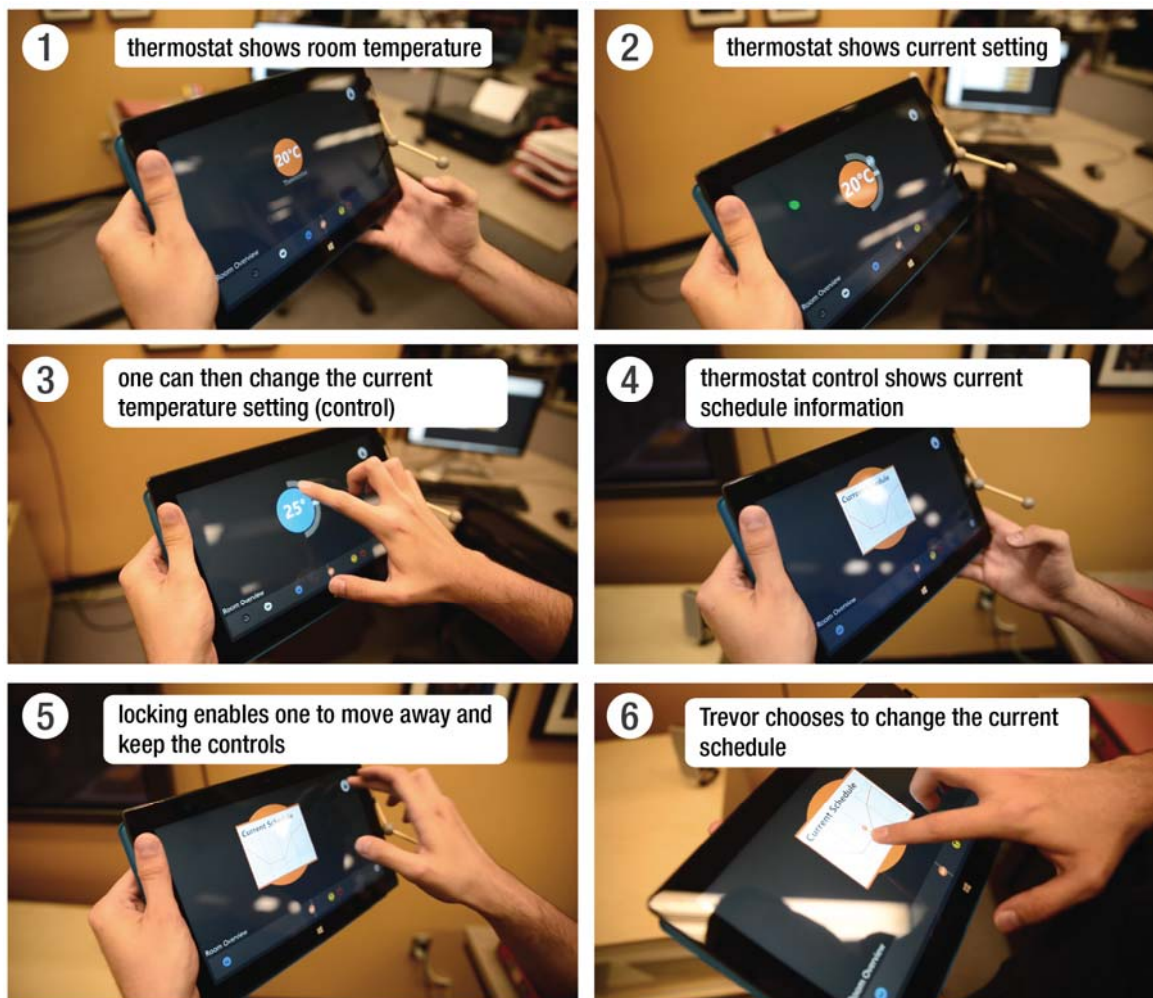


Figure 4.11 Main usage scenario 2 – Gradual Engagement.

temperature of the room (Figure 4.11-1). As he moves closer, he sees more detail and more opportunities to interact with the thermostat. He is able to see the current temperature setting (Figure 4.11-2) and as he continues to approach it, he can modify that setting (Figure 4.11-3). When he is very close to the thermostat, he sees the scheduling information (Figure 4.11-4). He looks at his heating schedule and decides to change it. He locks the screen so he can move his tablet around without losing content (Figure 4.11-5) and changes the schedule (Figure 4.11-6).

Discussion. This scenario realizes several of the concepts described in Chapter 3: gradual engagement (§3.2.2), zooming (§3.2.3.3), scheduling (§3.1.5), and basic locking (§3.2.2.1). More generally, this scenario illustrates how gradual engagement of controls work as a function of proximity. While this scenario focuses on a particular appliance (the thermostat), all other appliances implement this gradual engagement in a similar manner (as detailed in Figure 4.4 to Figure 4.8). By orienting his device to face an appliance, as described in the previous scenario (§4.1.1), Trevor was able to select an individual appliance within the ecology. His intent of interacting with it occurs implicitly by physically approaching it, where the interface uses a semantic zoom effect (§3.2.3.3) to show progressively more information of the thermostat state, and creates opportunities for interaction by revealing various controls. We already illustrated how the thermostat interface initially shows only the temperature of the room, followed by the temperature setting (Figure 4.4). This temperature setting is originally grayed out because it is initially displayed as a read-only value. When he walks towards the thermostat the setting value turns blue to indicate that it is now controllable and he is able to touch the control and change the current setting. Finally, as he gets very close to the thermostat the interface reveals the temperature schedule, which he is able to set freely (§3.1.5). If Trevor moves away from the thermostat, the process reverses, which we call gradual dis-engagement.

However, gradual engagement and disengagement—especially if done continuously—could interfere with interaction, as any movement towards or away from the appliance, or any shift in orientation, may change what appears on the

display. Consequently, the interface provides the capability to override gradual engagement by a 'locking' mechanism (§3.2.2.1), which freezes the interface to how it appears at this particular distance and orientation. In the scenario, we saw Trevor lock the screen before setting the schedule. While not strictly necessary, it means that, for example, Trevor can physically move or face away from the thermostat without changing the interface. Such movement would otherwise have been interpreted as a shift of focus.

4.2.3 Manually Overriding Proxemics

Scenario. Trevor is sitting on his couch watching a movie on the television. The lamp, which is several meters away, is overly bright so he decides to dim the lights.

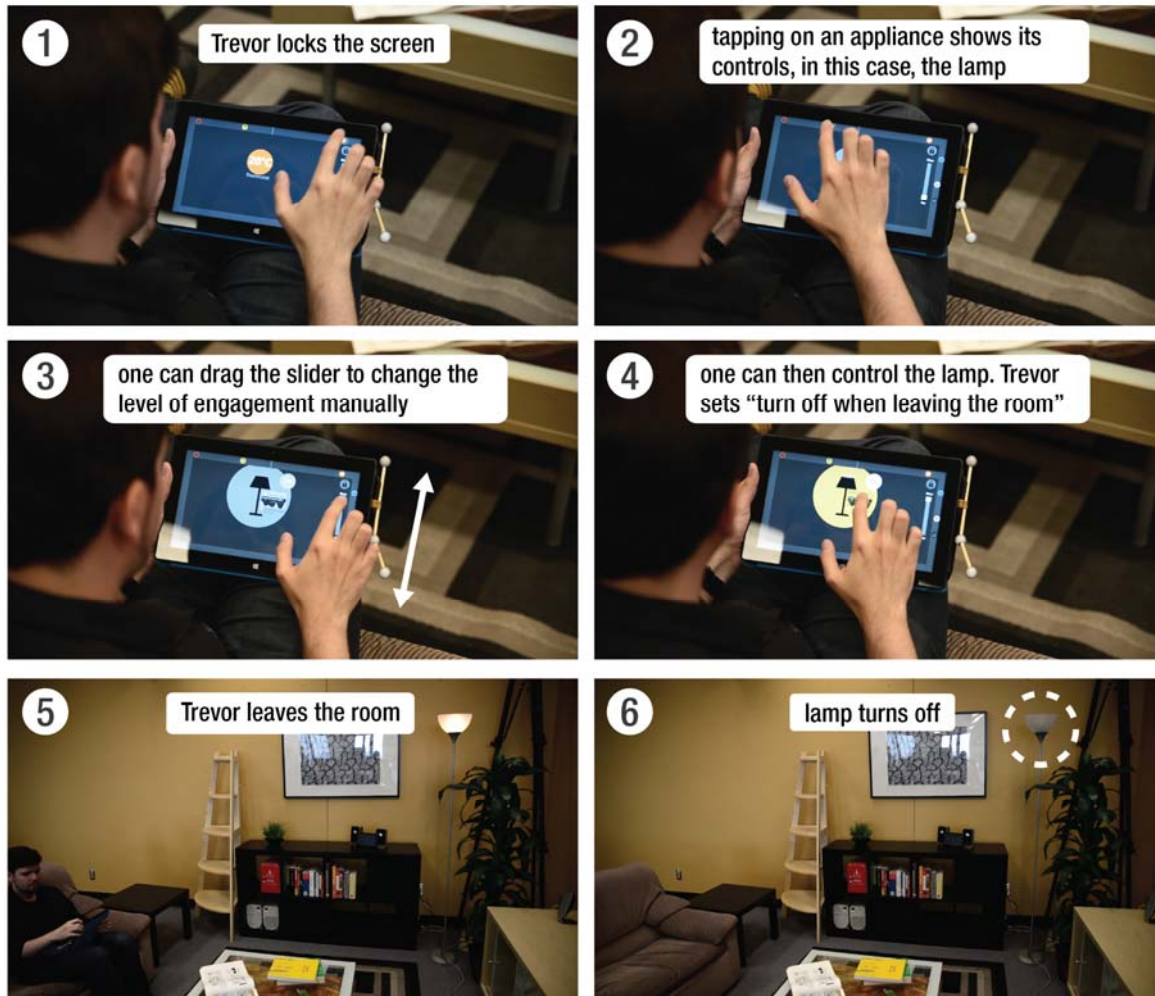


Figure 4.12 Main usage scenario 3 – Manual Override.

However, he doesn't want to get up. Therefore, he picks up his tablet (which happens to be oriented towards the television) and locks the interface, which enables him to tap icons and reveal specific controls. He first taps the thermostat interface (Figure 4.12-1). He then taps on the small lamp icon shown at the border (Figure 4.12-2), and the larger lamp icon appears at the center as if he had oriented the tablet towards it. The interface also reveals a 'proximity' slider (first introduced in Figure 4.3 on page 64), which allows Trevor to manually set the semantic zoom level (Figure 4.12-3)—by moving the slider, Trevor simulates the interface as though he has physically moved towards the lamp. Trevor drags the proximity slider for the lamp until he can view the brightness control (see Figure 4.5 for details). After he sets it to his desired level (Figure 4.12-4), he also sets the lamp to turn off when he leaves the room. After watching the movie, Trevor leaves the room, and the lamp turns off on its own (Figure 4.12-5, 6).

Discussion. This scenario illustrates two of the concepts described in Chapter 3: manual override and overview selection (§3.2.1), and shows another example of how basic locking can be used (§3.2.2.1). More generally, the scenario shows how the interface design should not insist on gradual engagement as the only interaction mechanism, and thus it affords other, more traditional mechanisms to choose between and navigate to controls. We saw that, unlike the previous scenario, Trevor decided to stay in one place rather than move towards an appliance (in this case the distant lamp), as doing so would have required extra effort and interrupted his movie viewing. Instead, he locks the interface and the selectable appliances at the tablet's border now serve as a graphical menu. As typical with menus, he taps to select a particular appliance. The interface shows the lamp interface at a medium semantic zoom level centered on the screen, i.e., as if he were standing close to it (less than a meter). A slider on the side lets him manually navigate to other lamp settings, otherwise accessible through physical proximity. As such, he can make the interface as simple or complex as he desires, and see settings appear or disappear as he looks for an appropriate proximity setting for the controls he wants to use. Alternately, instead of selecting the small

lamp icon of the lamp, Trevor could have oriented the tablet towards the lamp and then lock the interface. This selects the lamp through his initial orientation of the device, so that Trevor would now only have to adjust the slider. This shows how we can relax and incorporate different interaction approaches (scanning, touching and pointing) described earlier in Chapter 2 §2.2.3.

4.2.4 Around-Appliance Menu Navigations

Scenario. Trevor decides to set an alarm before going to bed; he approaches his radio alarm clock, and the tablet shows the radio icon interface (Figure 4.13-1, see also Figure 4.6 for details). When he is in close proximity (about 0.2 m away), he shifts his tablet to point slightly to the right of the radio; the interface animates to show a clock control (Figure 4.13-2). Using the clock control, he sets the alarm to the desired wake-up time. He then decides to play some music, while he's already interacting with the radio. To do so, he shifts the tablet slightly to the left of the radio. More detailed radio audio interface controls appear, and he presses play (Figure 4.13-3). Initially, the volume is too low, so Trevor re-orientes his tablets to face the speakers. This action brings up volume controls which he then adjusts accordingly. (Figure 4.13-4).

Discussion. This scenario emphasizes the spatially arranged controls and hierarchy concepts described in §3.1.4 and semantic zooming through micro-mobility in §3.2.3.3. More generally, the scenario illustrates how subsets of an appliance's controls can be virtually arranged around the appliance, where micro-mobility (via adjusting the tablet's orientation at a given proximity) is used to progressively reveal and navigate between those subsets. As a result, these interactions have been designed to take place in close proximity to the appliance.

The scenario illustrates two ways of associating this information spatially. The first one is to use *spatial references*, where the information connects to a virtual area around the appliance, such as controls situated above, below, to the left or to the right. In this example we use left and right to show two different types of controls. However, because these spatial references are abstract and must be learned, the

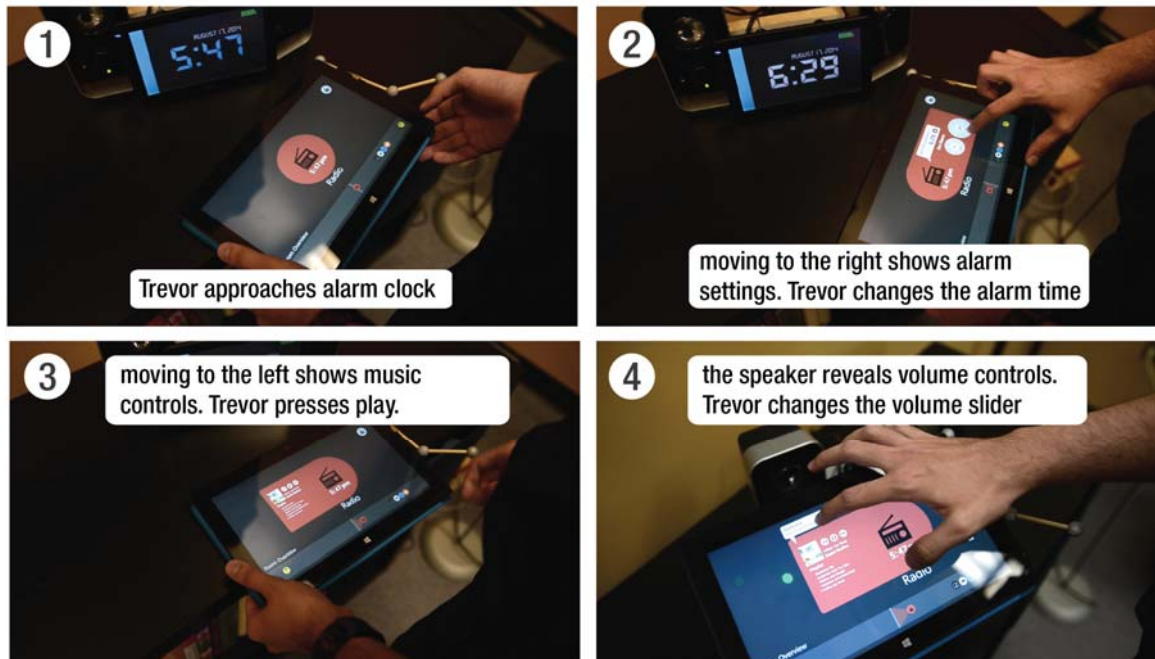


Figure 4.13 Main usage scenario 4 – Around-Appliance Menus.

interface could hint at their presence via (for example) arrows (i.e. a feedforward mechanism), which indicate that the person can shift the tablet in the indicated directions to reveal more content (note that we have not yet implemented these arrows). The second type of spatial association is through *semantics*, where specific parts of the appliance signify certain controls. In the radio example, the speakers are inherent to music volume, thus orienting the tablet towards the speakers reveals the volume control.

4.2.5 Room View Hierarchy

Scenario. Trevor enters his living room (Figure 4.14-1). At the entrance of the room, the interface shows an overview of the room and all its contained appliances; Trevor can see a few available controls (see Figure 4.9 on page 70 for details). From the entrance of the room, he turns on the TV (Figure 4.14-2, 3) and sits down to watch (Figure 4.14-4). Because he is now in the room as opposed to the entrance, the interface shows appliances individually, as discussed in prior sections.

Discussion. This scenario illustrates the appliance groups and hierarchies concept (§3.1.4). More generally, the scenario shows how appliances can be

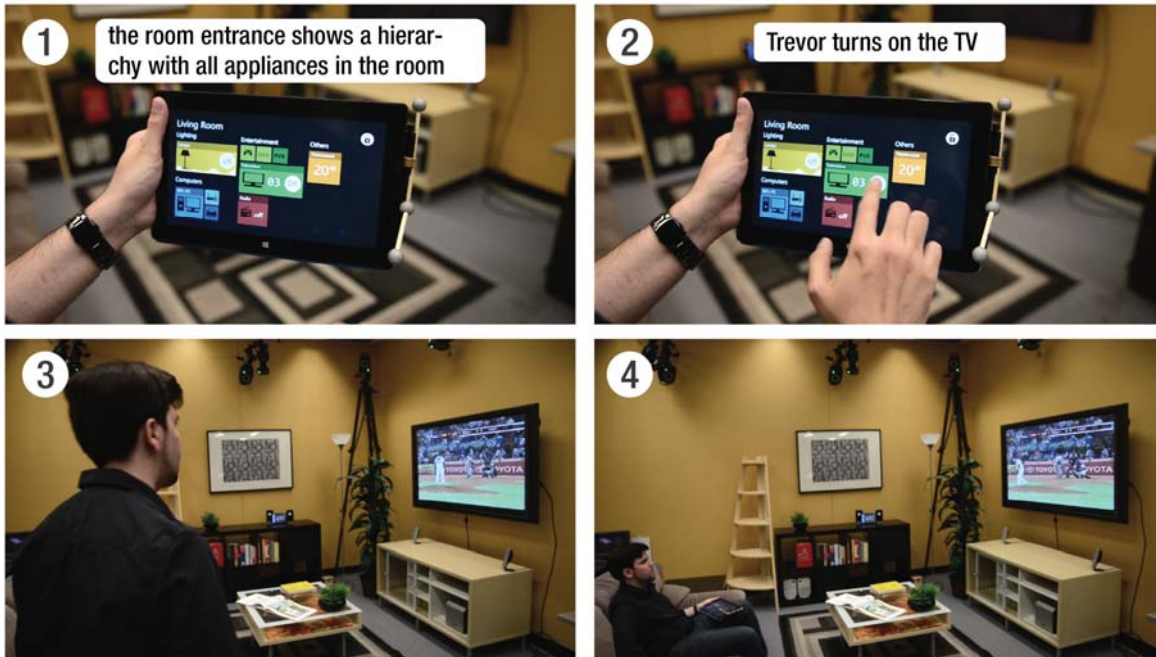


Figure 4.14 Main usage scenario 5 – RoomView Hierarchy.

grouped and arranged as a hierarchy, in which different levels of the hierarchy can be accessed as a function of proximity. Here, the room entrance serves as a fixed feature – a boundary – where the interface displays a high-level view of the contents of the room. Trevor can see all appliances that are in the room’s ecology and their primary settings. Trevor also has a small degree of control over several basic appliance functions, such as being able to switch the television on or off. This could be extended to reveal specific controls by tapping on a particular appliance, similar to what occurs when one taps on the small appliance icons shown at the borders (manual override, §4.2.3). The interface organizes and categorizes the different appliances in the room, and provides general information about them, such as showing the thermostat’s temperature, the state of the lights and the TV, etc. By locking the screen on this room view, he essentially is equipped with a control resembling a universal remote or a room console. Finally, the doorway can be considered a hierarchical overview activated by being located at the room’s threshold, where entering the room then allows implicit navigation (via proximity and orientation) to the next level of the hierarchy – the individual appliances.

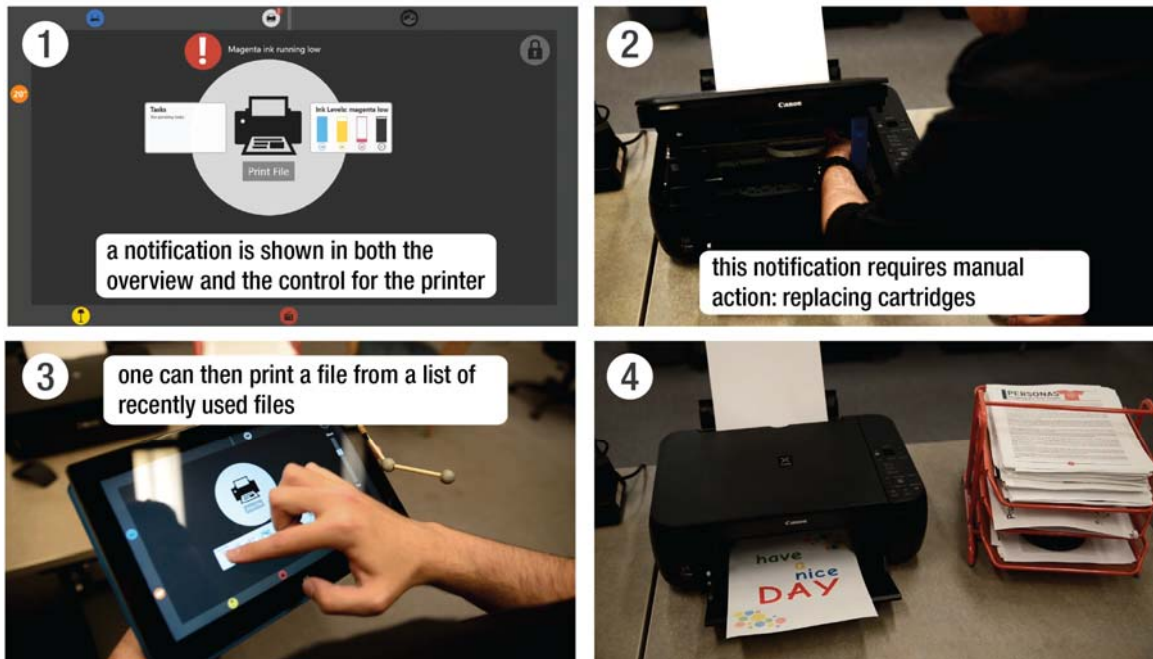


Figure 4.15 Main usage scenario 6 – Situating Context of Actions via Proxemics.

4.2.6 Situating Context of Actions via Proxemics

Scenario. Trevor wants to print a file. He sees on the overview that there is a red exclamation mark on the printer’s spatial reference icon. As he approaches his printer, he sees that there is a notification associated with it (Figure 4.15-1, see details in Figure 4.8) stating that the magenta ink cartridge is low. He goes to the printer and replaces the cartridge (Figure 4.15-2). When he picks up his tablet again, the notification has disappeared, confirming that the printer is working properly again.

He continues to print his file. While standing next to the printer, he taps on “Print File”, which reveals a list of recent files (Figure 4.15-3, see details in Figure 4.8). He selects “Card.pdf”, which sends that file to the printer, and retrieves his printout (Figure 4.15-4).

Discussion. This scenario illustrates the concepts of how proxemics can serve to situate the context of actions (§3.2) and how an appliance can leverage notifications and file transfers (§3.1.5). The printer ambiently communicates that it has run out of ink by several means. First, it flashes an exclamation mark on the

visualization overview icon so that he knows something is wrong with the printer even if he is standing away from it. He also knows which printer is out of ink (if there are several printers), because the overview indicates the physical direction of that printer. Second, Trevor is able to view more information about that alert – in this case that the printer is out of magenta ink – as he approaches it. At this point, he can choose to do nothing (perhaps also choosing to hide the notification) or address the problem and replace the cartridge.

The next part of the scenario demonstrates how a print destination can be selected simply by standing next to it, i.e., the usual print dialog asking the user to select a printer is not required. Trevor implicitly selected the desired printer by approaching it; all he needs to do is select the file to print. This illustrates an alternate way of performing a common task that already requires physically approaching a device (approaching the printer to retrieve the printout) and to remove the context switch of interacting with a separate device (e.g. desktop computer), navigating through a file system, finding a file, opening it, and setting it to print and then physically approaching the printer to grab the printout.

4.3 Secondary Scenarios

This section turns to additional scenarios that demonstrate other concepts that – while not central to our thesis of gradual engagement via proxemics – demonstrate other ideas that build on our foundations established in Chapter 3.

4.3.1 Identity-Based Access Levels

Scenario. Tina, a guest in Trevor's house, wants to increase the temperature setting of the thermostat (Figure 4.16-1,2). However, she finds that she is unable to change the current setting (Figure 4.16-3). The reason for this is that Trevor—who is conscientious about reducing his energy use—has configured the system so that only he is able to change the thermostat state.

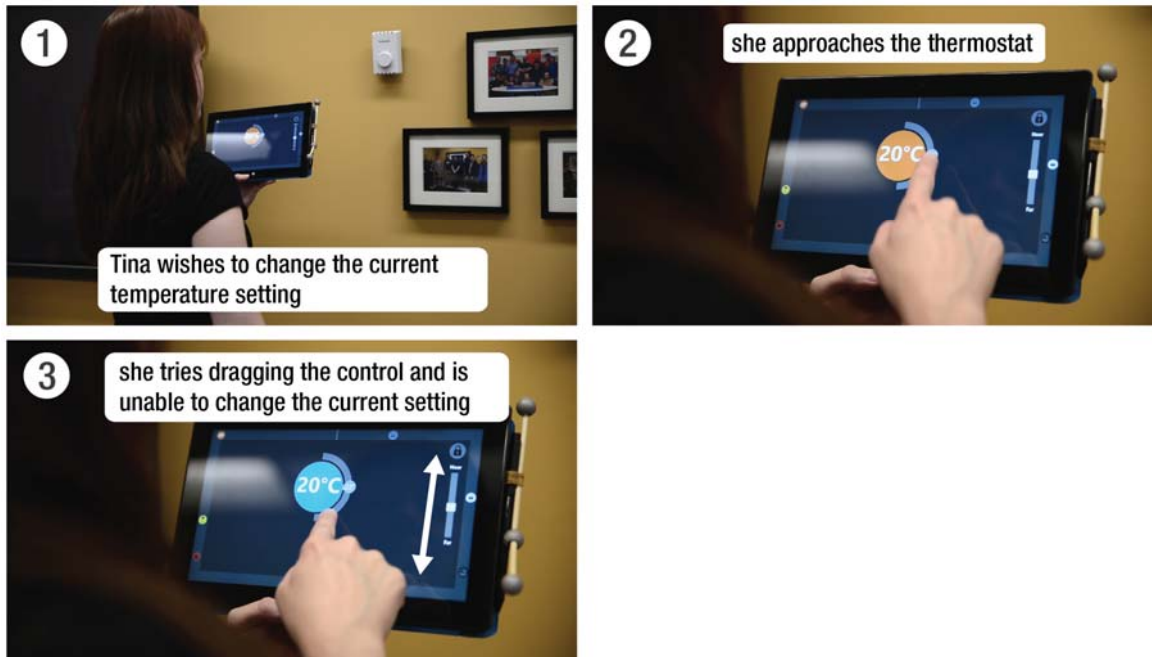


Figure 4.16 Secondary usage scenario 1 – Appliance Access Levels.

Discussion. This scenario illustrates the identity and access levels concept (§3.2.1). More generally, this scenario exemplifies how we can leverage an individual’s identity to restrict controls, similar to parental controls but without requiring a password entry. This particular case shows how when an unauthorized guest tries to control the thermostat, they are restricted from controlling it. This adds a layer of security to our system. Furthermore we could think of other rules that can be established, such as allowing Tina (the guest) to change the temperature only if Trevor (the home owner) is present. Such an arrangement builds upon traditional social conventions, where people use their own interactions to mediate what the other can do.

4.3.2 Extending Appliance Functionality and Leveraging Existing Controls

Scenario. Trevor tries to go online and finds that his internet connection is slow. He approaches his router to view its settings and bandwidth use in more detail (Figure 4.17-1, see also Figure 4.7 for details). He sees devices that are currently connected to that router and finds that his PC to the left is currently connected and

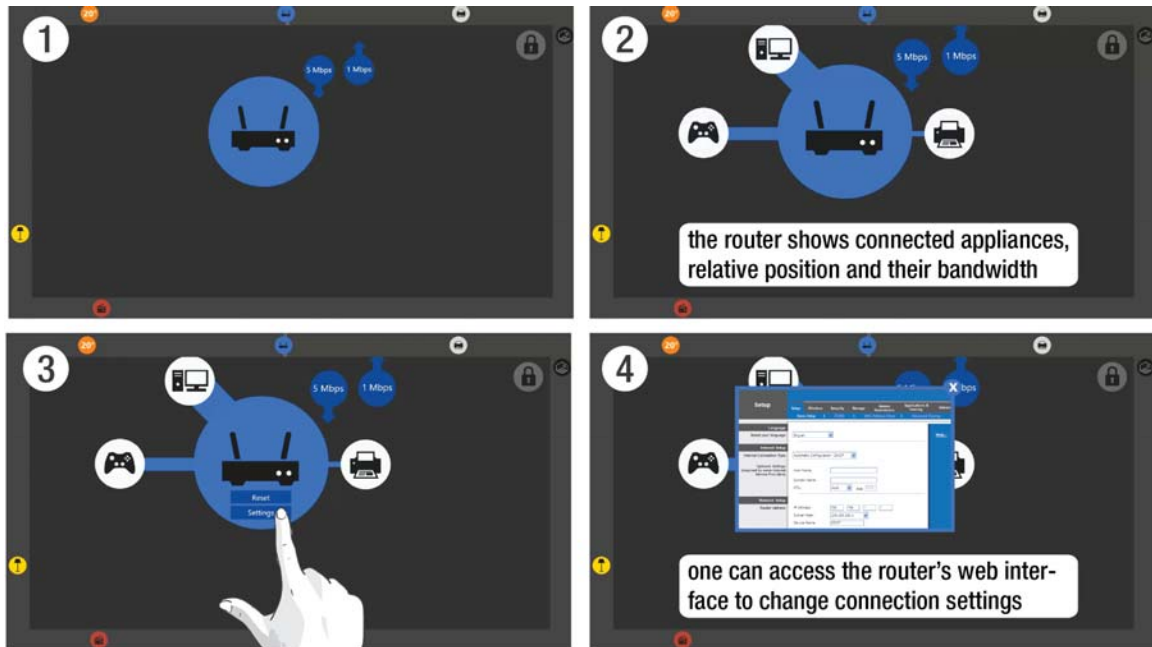


Figure 4.17 Secondary usage scenario 2 – Extending an Appliance Functionality. consuming a lot of bandwidth (Figure 4.17-2). He goes to his router settings (Figure 4.17-3), where a traditional router interface on a web browser integrated into the controls and restricts the connection speed for his PC (Figure 4.17-4).

Discussion. This scenario shows how we can extend appliance functionality through a centralized system, where we leverage existing interfaces¹. In this case, a centralized system can extend an appliance’s functionality. Thus, an appliance that is typically not able to reveal certain information can leverage information provided by a server. The router knows the IP addresses of currently connected devices, but has no idea what those devices are. However, the central server controlling the ecology can look up the identity of the devices associated to these IP addresses. That is, the ecology as a whole can integrate information from throughout the system to extend the capabilities of individual appliances in that space. Somewhat similar integration is now appearing in other technology, such as

¹ The router interface we built is not fully functional. That is, while it reacts to gradual engagement as shown, the connected devices and web interface is a mockup.

Chapter 4 · Proxemic-Aware Controls in Action

a TV leveraging information from a cable box to reveal schedules and search functionality or personal video recording.

The scenario also introduces a second concept: the integration of existing interfaces into our system. Router interfaces are typically created and accessed as a web portal with all of the settings and controls required to set connection passwords, enable ports, perform firewall protection, etc. Here, the appliance interface redirects the person to the existing router interface. We saw how Trevor is redirected to the web portal when he taps on the settings button. In designing the system, we could have instead created a new interface that fetches all the router's web information, and presents it in a manner that is visually consistent with other appliance and perhaps better leverages gradual engagement. However, it is only possible if the appliance allows access to its internal settings (perhaps via proprietary remote procedure calls). This adaptation is expensive, and would likely need be done on a per-model basis as different models of the same appliance would have different settings and ways to access those settings. By hooking into existing interfaces, potentially any brand of router could be used without worrying about compatibility issues.

These different scenarios have shown how the concepts presented in the proxemics-aware controls conceptual framework can be applied to the design of remote controls for ubicomp ecologies. The next chapter focuses on the system details: the technical setup, the development process, the system architecture and the limitations of the system.

Chapter 5. System Details

The previous chapter outlined a set of scenarios that demonstrated implemented concepts pertaining to how *we can use socially-established protocols (proxemics) to seamlessly interact with an ecology of appliances*. This chapter explores and adds the implementation details that made these scenarios possible. First, we broadly examine our technical setup (§5.1). Next, we describe decisions made during the development process on how to present the interface – showing objects in space and transitioning between different levels of proximity (§5.2). We also reflect on our refinements from previous iterations¹. Next, we delve into our system architecture, where we describe our software solution and implementation details behind our prototype (§5.3). Finally, we expose some conceptual limitations and technical constraints of our system (§5.4).

5.1 Prototyping Environment

This section aims to broadly describe our technical setup. It serves as a way to contextualize our system implementation and enable further discussion in the next sections of this chapter, particularly §5.3.

Our *Proxemic-Aware Controls* system, which is set up in a room in a research lab that simulates a home environment, is known as the “Home Space”. Its general

¹ Portions of this Chapter have been published as:

Ledo, D. and Greenberg, S (2013) - Mobile Proxemic Awareness and Control: Exploring the Design Space for Interaction with a Single Appliance. In Proc. ACM CHI 2013 Video Program (Juried) - CHI'13. (Paris, France), 1 page abstract plus video, April 27 - May 3.

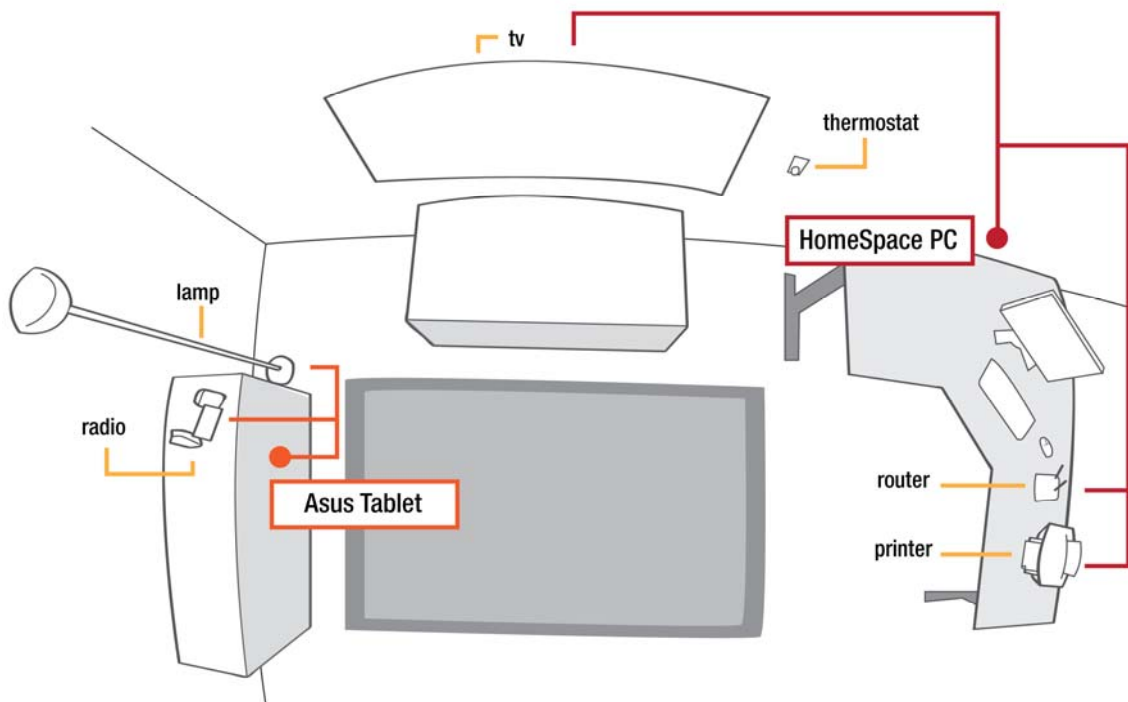


Figure 5.1 Technical setup of our implementation. It showcases the location of the individual appliances and which computer they were connected to in order to operate. layout is illustrated in Figure 5.1. The Home Space contains various furnishing, e.g., a couch, shelves, side-tables, as well as a large display, which is used as the television appliance. We placed five additional physical appliances in the Home Space: a thermostat, a lamp, a router, a printer and a radio alarm clock, all located as shown in Figure 5.1. Some of our appliances are interactive, either fully or partially under digital control. This means that a person is able to change their state via the network. For example, changing the lamp’s brightness level or turning it on or off affects the lamp’s physical state. Playing music on the radio, setting the volume or changing the alarm time on our interface are also reflected on the radio itself, which can be seen on the radio’s display or heard through the speakers. The television can be turned on or off. Other appliances, such as the thermostat, the router and the printer are not actually connected to the system. Instead, we use them as place holders with mockup interfaces which are fully interactive on the mobile device performing the control, but the changes are not actually reflected on the physical appliances. As a result, individuals are able to modify their virtual

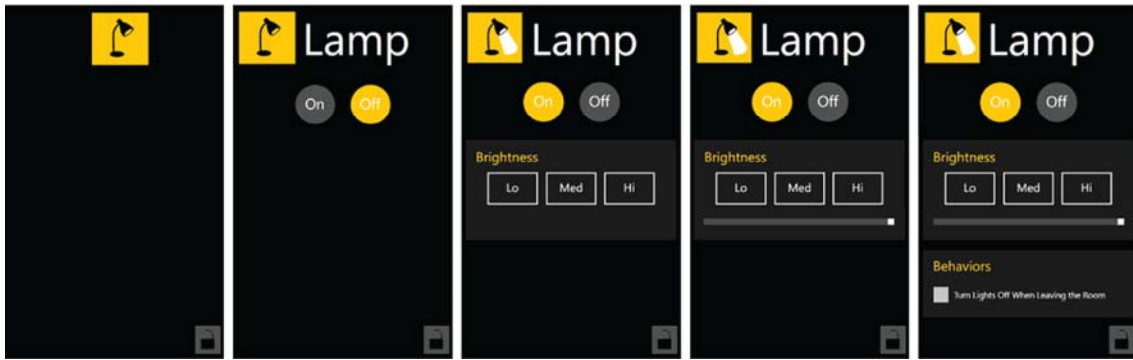


Figure 5.2 First iteration of our lamp interface showing the discreet progression of stages as a function of proximity.

state and see the changes reflected on the server, but those changes do not affect the appliance. §5.3.2.1 discusses the room’s tracking equipment.

5.2 Visual Representation

In §4.2 and §4.3, we showcased different usage scenarios for Proxemic Aware Controls. Most of these scenarios illustrated two key points on how we implemented the gradual engagement pattern to remotely control appliances. These are: (1) changing the interface as a function of distance and (2) positioning different controls on screen as a function of orientation and distance. The following two subsections explain these two points in further detail.

5.2.1 Changing the Interface as a Function of Proximity:

Discrete versus Continuous Engagement

As discussed in §3.2 engagement is described as a function of distance, we described engagement as a function of distance, which could be either discrete (different amounts of content are revealed or hidden when distance thresholds are met) or continuous (engagement is directly a function of distance).

Our first design iteration made use of discrete stages – information would be revealed when certain distance thresholds were met. This can be seen in Figure 5.2, which shows how our initial interface expands as different thresholds are crossed:

first showing presence, then the name of the appliance, then basic controls followed by specific controls (e.g., lamp brightness levels), then with a light dimmer and finally with behavioral settings (e.g., the lamp turns off when the user leaves the room). What these still images do not show is that the interface between each stage remains static until the individual crosses a distance threshold; this is better illustrated in our CHI Video (Ledo & Greenberg, 2013). Once the person crosses a threshold, an animated transition plays revealing additional content for the interface. To prevent drastic interface changes between transitions, we followed a layered approach. That is, as Figure 5.2 shows, extra content is stacked in downward direction as people approached the appliance, whereas prior elements tend to remain in their original position. Thus, the user is able to navigate towards increasing levels of flexibility and complexity for the interface without losing or seeing major changes to the content from the previous state.

However, after testing our prototype we found that the system did not feel particularly responsive; we found it more appealing to reveal items continuously as a function of distance. We implemented this approach in our new tablet version, demonstrated in the scenarios in §4.2 and §4.3. We believe that this continuous change, particularly taking place with *the zooming of the interfaces as a function of proximity*, leads to a richer and smoother experience. We describe how we achieved this continuous transition later in this chapter.

5.2.2 Transitioning Between Controls as a Function of Orientation and Distance: Showing Objects in Space

Our first iteration focused on the interfaces for single appliances, rather than considering the ecology as a whole. A major challenge we faced in our second iteration was how to portray multiple appliances and switch between them. We had to reveal the currently engaged appliance on screen, and further consider how one would transition from one appliance to another. To do this, we consider engagement as a function of orientation and distance. Our basic approach is the following: *when directly facing an appliance (the angle between an appliance and*

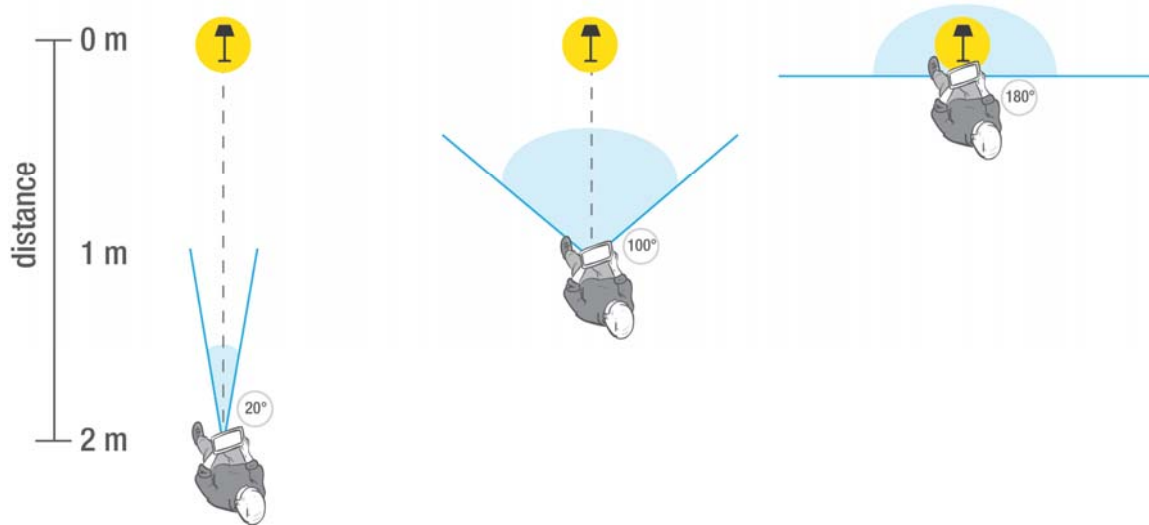


Figure 5.3 Field of view for an appliance as a function of distance: the angle threshold that establishes engagement increases as one approaches an appliance.

the tablet being 0°), the orientation engagement is the highest. If the tablet is moved to the right, the interface control moves away from the tablet's field of view as the object is to the left of the tablet. Similarly, moving the tablet to the right will make the appliance be at the left of the tablet.

We extend the notion of engagement through orientation by looking at distance to establish an orientation threshold, as it provides us with more control for what is shown on screen: *if one is near an appliance, then one needs to turn further away from it in order to disengage with it.* Conversely, *if one is far away from an appliance, the orientation threshold required to establish engagement is smaller,* so smaller changes in orientation of the tablet lead to disengagement at a higher rate. We believe this balances and encourages navigation and focus: a person can quickly scroll through different far away objects to become aware of their presence and walk towards it to increase their engagement with it. This effect, added to how we use distance to determine the size of the controls (i.e. objects far away are smaller, and as one approaches them they grow in size) encourages the notion that one is indeed engaging with the desired appliances.

Our actual implementation reflects this notion: when a person is 2 meters away from the appliance, the angle required to show an appliance on screen is

established to be within 20° (10 left, 10 right), which captures a narrow field of view (Figure 5.3 left). When directly in front of the appliance (0 millimeters away), the engagement angle threshold is 180° (90 left, 90 right); therefore, in order to completely disengage with the appliance, one would have to turn at least perpendicular to the appliance (Figure 5.3 right). Our engagement angle threshold is determined as a linear interpolation between these two distances (2 m to 0 m). Our Figure 5.3 reflects this, as being 1 m away finds an orientation range interpolated between 20° for 2 m and 180° for 0 m, resulting in 100° . Thus the closer the tablet is to an appliance, the further away they have to face in order to disengage with it. The farther away the tablet is from an appliance, the easier it is to disengage with it. An example of this is to scan the appliances present in the room by a sweeping motion. For description purposes, we refer to this distance-based orientation range as $-\alpha$ (the left-most side), $\alpha=0^\circ$ (centered) and $+\alpha$ (the right-most side). We selected our start and end positions (0-2 m) and our angle thresholds corresponding to them (20 - 180°) by manually testing different thresholds in the room and finding that these numbers worked best. While there may be other mathematical models that can consider the room dimensions and the distance and orientation thresholds, they are beyond the scope of this thesis.

5.2.2.1. Positioning Objects on Screen

The next step was to present the interfaces on screen. The dynamism of the interfaces meant that as a person disengaged from an appliance by changing their orientation, the interface should update accordingly. We developed and tried three different approaches: transparency, horizontal positioning and a hybrid of both. These techniques are shown in Figure 5.4, in which each column corresponds to different orientations of the mobile device with respect to the appliance: left side ($-\alpha$), directly pointing at the appliance ($\alpha=0^\circ$) and right side ($+\alpha$).

Transparency. Our first approach, explored in our first iteration, used transparency as the means to determine engagement (Figure 5.4 Top). If the tablet is directly facing the appliance ($\alpha=0^\circ$) then the control for the appliance is fully

opaque; and when the tablet is facing away from the appliance, the opacity changes to make the control fully transparent. While transparency served as a good mechanism to illustrate the level of engagement, we found that areas other than the center of the screen are left completely unused, and that when multiple objects were within the field of view as a result of overlapping α ranges, only one of them would be visible. Transparency also proved a poor way of indicating the angle that the person had with the appliance increasing the difficulty to spatially associate the control with the physical appliance.

Horizontal Positioning. Our second approach used horizontal positioning (Figure 5.4 middle). Here, the α orientation value is mapped to the horizontal position of the appliance control icon on the screen. If the tablet is facing towards the appliance ($\alpha=0^\circ$) then the icon would be placed in the center of the screen. As one turns away from the appliance, the control would move towards the edges of the screen (i.e., reflecting its position in the field of view), eventually disappearing.

While horizontal positioning created a sense of navigation within the interface, it has a significant problem: small movements of the tablet generate large movements of the icon position, which complicated the user experience. While a person can still see the icon's information, these large movements became more of an issue when people tried to perform direct touch actions to control the appliance, such as dragging a slider. This led people to often and easily miss the targets they wished to interact with.

Hybrid of Positioning and Transparency. We built on the strengths of both previous techniques by combining them. In our initial version, the icon would move completely across the display, where it was just transparent at $\pm\alpha$ and fully opaque at ($\alpha=0^\circ$) and the controls moved across the edges of the display. However, we found that, while we did improve the overall feeling of the interface, it was still possible for controls to move accidentally across the display when small movements on the tablet were performed. To mitigate this, we decided to restrict this motion to stay within the middle third of the screen (Figure 5.4 bottom). We

also added software to correct for large side-to-side movements, so that the icon is generally positioned close to the center (see next section). As a result, the target motion was much more stable. Ultimately, horizontal motion combined with transparency proved a suitable cue.

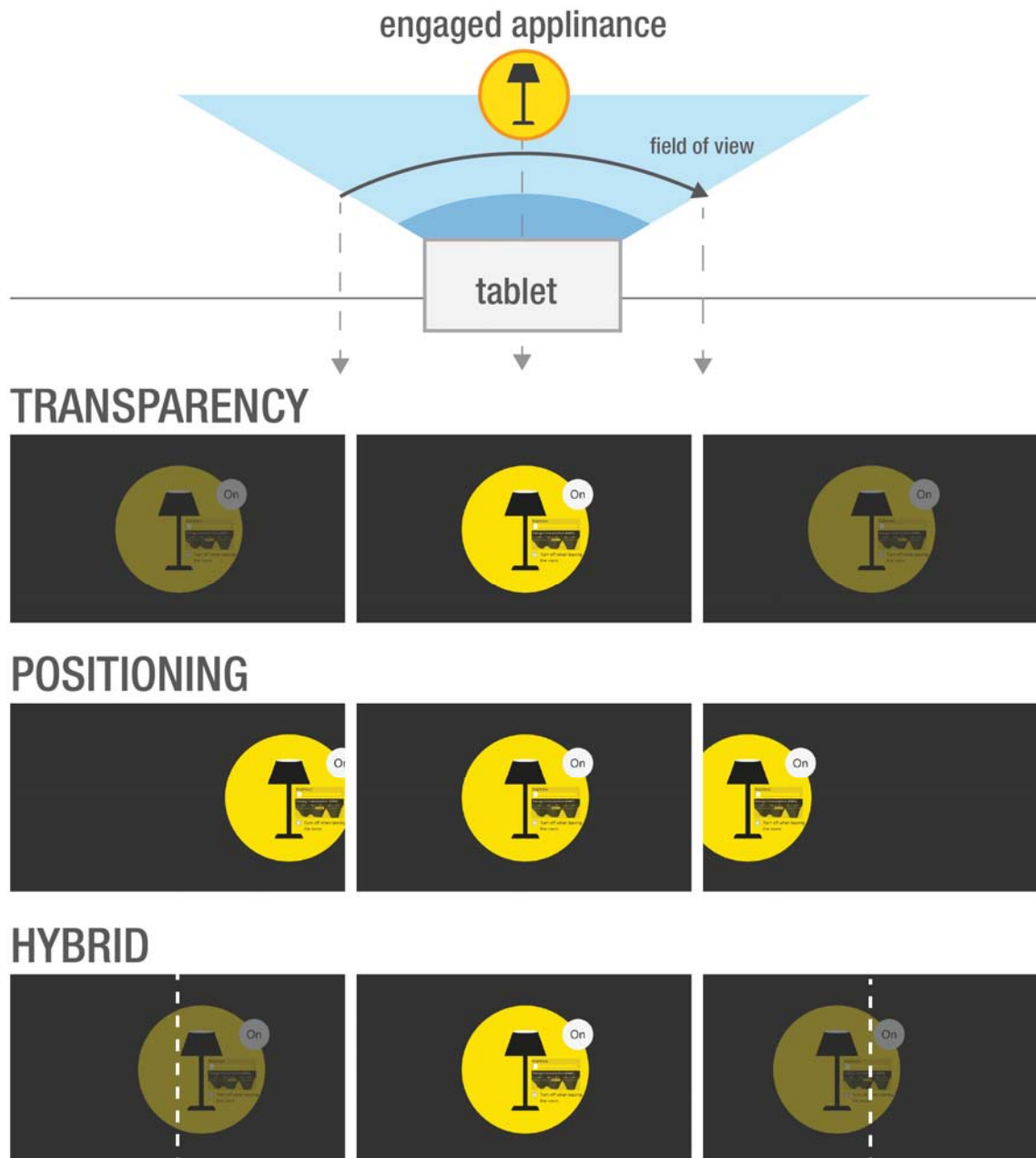


Figure 5.4 Object positioning on the screen as a function of orientation. This shows how we followed three different approaches: transparency, positioning and a hybrid that incorporated both approaches (added dotted line shows off-centered positioning).

5.2.2.2. Correcting Transitions

Both of our approaches – horizontal positioning and transparency – were problematic when done as a strictly linear mapping between the tablet’s appliance icon orientation and the screen position of the control. As shown in Figure 5.5 (left), linear mapping meant that, for our transparency technique, the only time when controls are fully opaque (and thus clearly readable) was when the tablet was directly or almost directly facing the appliance ($\alpha \sim 0^\circ$). For the horizontal positioning, small changes in orientation could heavily impact the position of the control, making it very difficult to visually track and to interact with (i.e. touch) interface controls. As a result, we applied a cubic mapping function, illustrated in

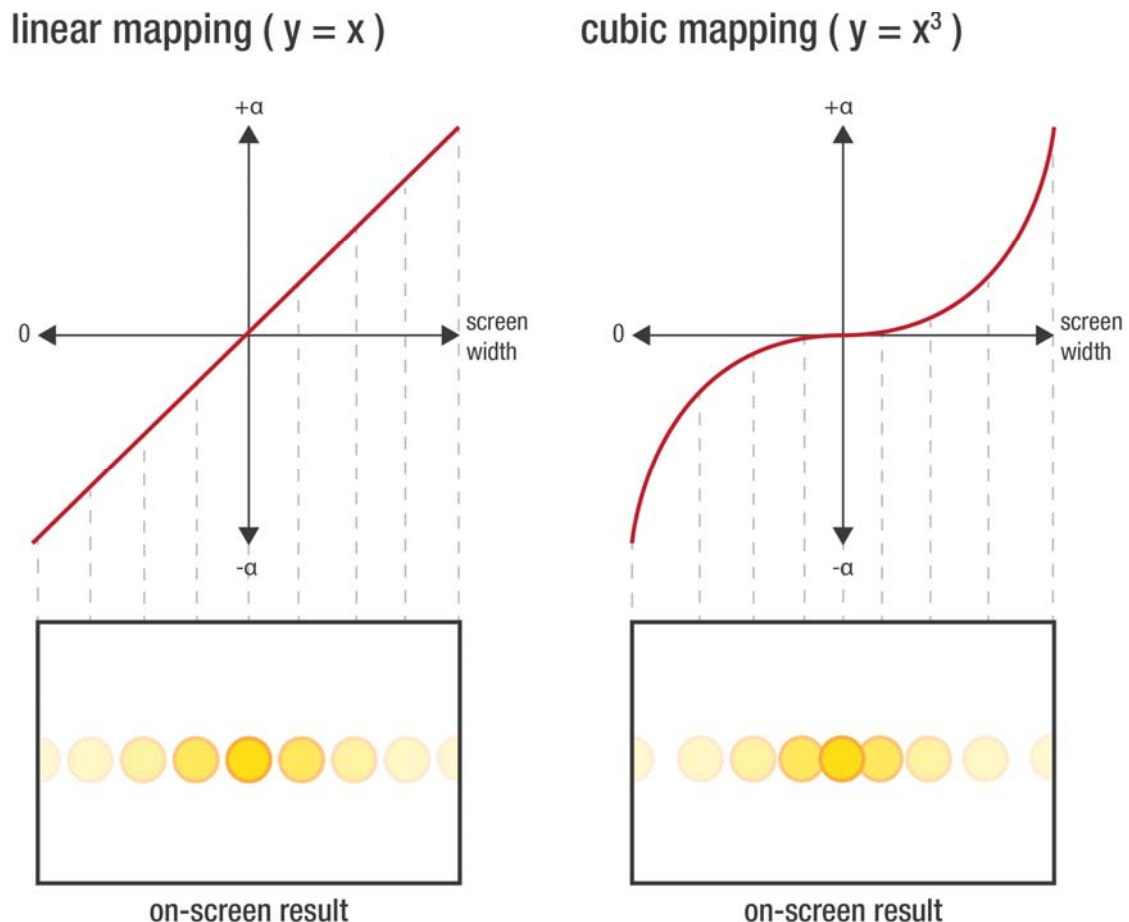


Figure 5.5 Our original method for mapping orientation to screen positioning (linear mapping) compared to our correction mechanism (cubic mapping).

Figure 5.5 (right). The correction algorithm that implements Figure 5.5 works as follows.

1. Normalize angle from $-\alpha$ to $+\alpha$ to -1 to 1 through linear interpolation.
2. Apply function f to the current orientation value. For linear mapping, $f(x) = x$. For our cubic mapping, $f(x) = x^3$ where x ranges from -1 to 1.
3. Map the value resulting from $f(x)$ to a range from 0 to the screen width through linear interpolation.

The linear interpolation function that we apply is:

```
double Map(double value, double min, double max, double newMin, double newMax)
{   return (((newMax - newMin) / (max - min)) * (value - min)) + newMin; }
```

We also tried other odd power functions (such as $f(x) = x^5$). However, we found that these functions made the transition too quick and sudden.

We tried this correction on all three techniques – position, transparency and hybrid approaches. Overall, we believe that cubic mapping on the hybrid approach largely improves the interactive experience, and creates a transition that benefitted both transparency and positioning.

5.3 System Architecture

We now elaborate on the technical setup described in §5.1. Our system components are: networking, tracking, appliances, and software operated by the devices. We explore the general system details and the networking and tracking information. We then explain how we implemented our appliances, and finally we describe the software architecture of our system.

5.3.1 General System Details

Our system was set up in the Home Space. The system tracked the position and orientation of a mobile device in that space, and knew where particular appliances were located.

5.3.1.1. Networking

A Vicon motion capturing system tracked the position and orientation of our devices, and runs it through our server machine (Windows 7, 2.4 GHz Intel Core2 Quad processor). We use the Proximity Toolkit (Marquardt, Diaz-Marino, Boring, & Greenberg, 2011), developed in our lab, to gather tracking information from the Vicon system, translate that information into proxemic relations between tracked entities, and deliver that information to our server. This proximity information arrives at our server which delivers it to the corresponding mobile devices. In order to connect the tablet and phone to the server, we used the iNetwork² networking toolkit, also developed in our lab. Our first version of the client mobile devices made use of a Nokia Lumia 800 with Windows Phone 7 (where we developed its software using SilverLight and C#). Our second prototype used a Surface Pro 2 (using WPF and C#). In both cases, the mobile devices interacted with our system server i.e., a client/server architecture.

5.3.1.2. Tracking

To track locations of appliances, we originally used trackable Vicon markers similar to those used to track the mobile devices. However, we found that the current server computer was not able to handle more than four marker sets. To mitigate this problem, we created virtual entity locations (3 dimensional coordinates) that corresponded to the position of the appliances and integrated these locations into the Proximity Toolkit information. As we will see soon, these coordinates were stored in the appliance models. While we did resort to using virtual appliance locations, our architecture easily accommodates real-time tracking of appliances via markers, albeit with performance constraints.

We tried two different methods to track the tablet's orientation. The first method used vectors provided by the Proximity Toolkit, which define the tablet's up and forward directions. However, we found that the values given by the toolkit were

² <http://grouplab.cpsc.ucalgary.ca/cookbook/index.php/Toolkits/INetwork>

not smooth due to networking limitations. Furthermore, we also found it difficult to configure the motion tracking cameras to accurately capture small movements in the large area encompassing the room. As a result, even small inaccuracies could result in large changes in the interface. Our second method used the tablet's built-in compass and inclinometer to detect orientation. This produced smoother and more accurate tracking of the tablet's orientation, which sufficed for our final prototype. However, the compass required frequent recalibration. This could be mitigated in future versions by using the proximity toolkit's vectors at certain time intervals to verify and recalibrate the compass readings.

5.3.2 Appliances

We made use of six different appliances, as shown in Figure 5.6 and as described in the previous chapter. They include: a thermostat (Figure 5.6 a), a lamp (Figure 5.6 b), a radio alarm clock (Figure 5.6 c), a television (Figure 5.6 d), a router (Figure 5.6 e) and a printer (Figure 5.6 f). Three of these appliances were fully controllable (TV, lamp, radio) while the other three acted as placeholders with fully interactive interfaces on the mobile device (thermostat, router and printer). To implement control of the TV, we used a large display connected to the server machine. This

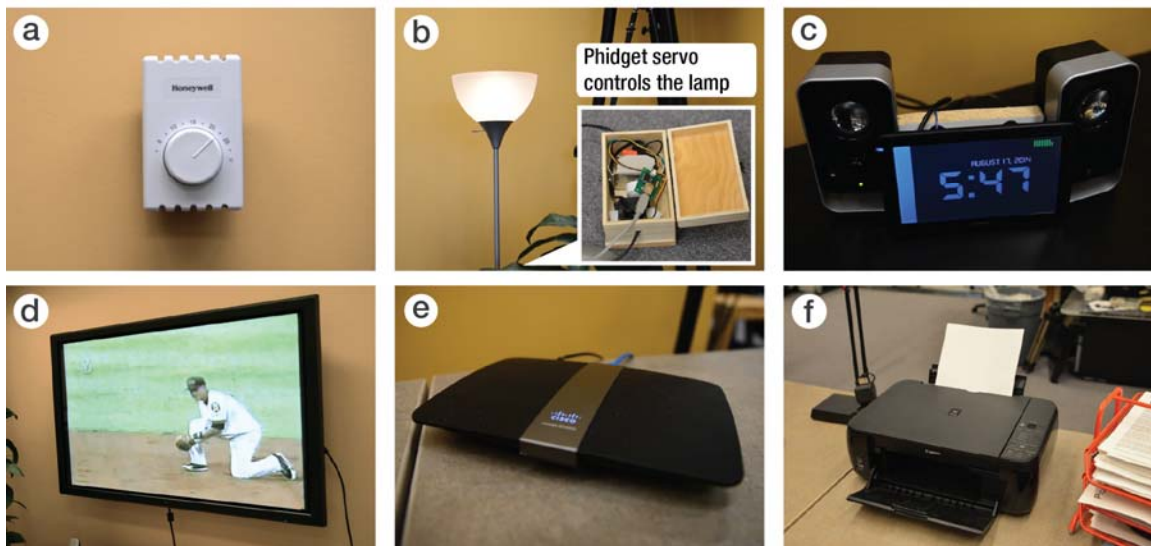


Figure 5.6 Implemented appliances featuring: (a) a thermostat; (b) a lamp; (c) a radio alarm clock; (d) a television; (e) a router and (f) a printer.

display plays videos when turned on and stops them when turned off. To control the lamp, we modified its circuitry (see Figure 5.6 b): we added a dimmer switch attached to a servo motor, controlled through Phidgets (Greenberg & Fitchett, 2001). The servo controller connected to an Asus EP121 Tablet PC, which communicated with our server. This Asus Tablet PC was a dedicated client that served to control both the lamp and the radio (shown in Figure 5.1 on page 88). Our radio was implemented by using a NIMO picture frame, which is a small display that we attached to our Asus Tablet PC. To play music, we connected speakers to the Asus Tablet PC. Each prototype appliances is a proof of concept demonstrating that individuals could interact with such appliances; they are not intended as finished products.

5.3.3 Underlying Software Structure

Our system relies on communication between four main programs (shown in Figure 5.7): *NetworkVisualizer* (server), *ProximityViewer* (the tablet

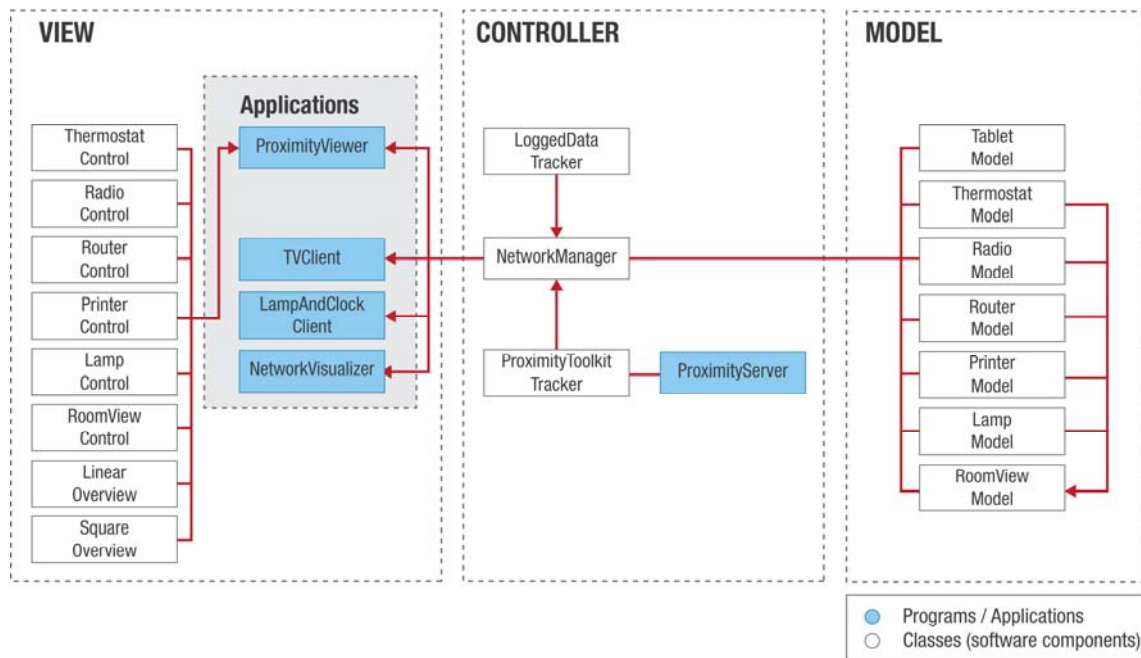


Figure 5.7 System architecture. Our architecture follows a distributed Model-View-Controller design pattern, which makes it so that a centralized server controls the models for each individual appliance. The mobile device receives data from the models and renders the content accordingly.

application shown throughout this thesis), `TVClient` (program that operates the large display) and `LampAndAlarmClockClient` (program that operates the radio and lamp). Furthermore, we use the Proximity Toolkit's *ProximityServer* to load the tracking information of the entities in the room. We run a single instance of each of these applications in different machines: the user's tablet runs `ProximityViewer`, the HomeSpace PC executes the `NetworkVisualizer` and the `ProximityServer`, as well as the `TVClient`, while our Asus Tablet PC executes the `LampAndAlarmClockClient`. The appliance distribution and the computers they are connected to is shown in Figure 5.1.

5.3.3.1. Running our System

To run our system, we first start an instance of the Proximity Toolkit server (`ProximityServer`). This is an application that is installed in our prototyping computer (HomeSpace PC in Figure 5.1). Next, we run an instance of `NetworkVisualizer`, a program that initializes our `NetworkManager`. This manager is a component that fetches all the information from the Proximity Toolkit (i.e. the position and orientation of the mobile device and any other tracked entities if any) through our `ProximityTracker` class, loads the locations for the appliances (hard coded or from a text file), and initializes all the models for each appliance. From here, `NetworkVisualizer` (shown in Figure 5.8) acts as our server to all other client applications. `NetworkVisualizer` also helpful for debugging, as one can examine what the system currently sees and is executing. Figure 5.7 shows all of these classes and their interactions, and it emphasizes that instances of `NetworkManager` are present in each running program with the goal of keeping track of the current state of all appliances and devices.

To operate the appliances, we run instances of `TVClient` on our HomeSpace PC (which is connected to our large display, as shown in Figure 5.1 on page 88), and `LampAndClockClient` on our Asus Tablet PC (which is connected to our lamp and our radio, see Figure 5.1 for details). To run the mobile/tablet application that controls them (showcased throughout Chapter 4), one executes

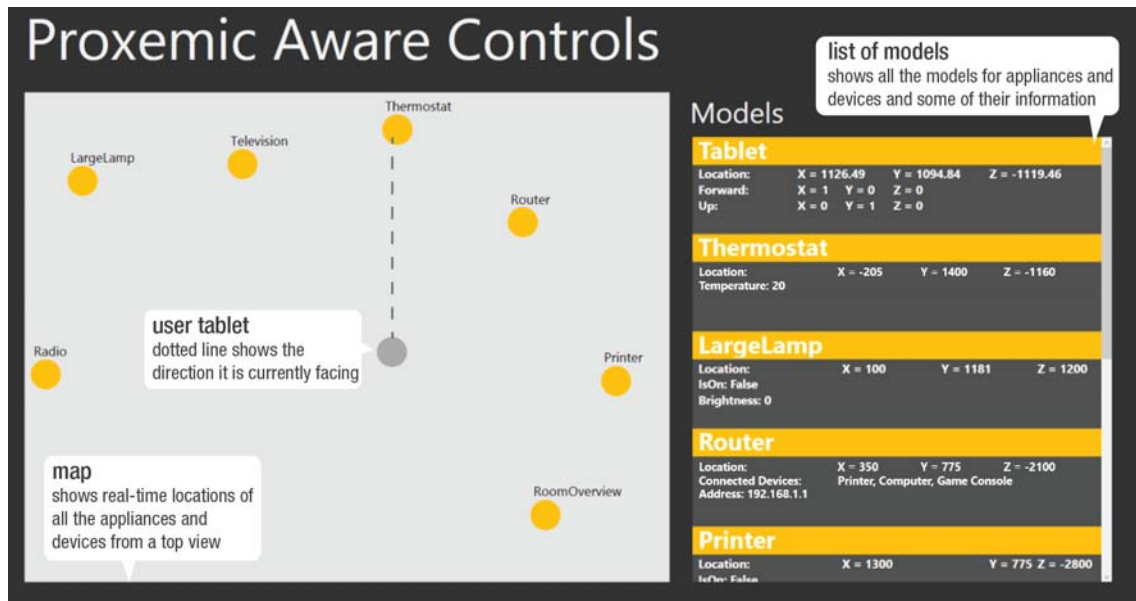


Figure 5.8 Screenshot of our NetworkVisualizer application. On the left one can see the different connected appliances and devices in a map that updates dynamically. To the left one can see all the models along with some of their information.

ProximityViewer. This is the only application our users are aware of operating throughout their experience with our system. Changing the location and orientation of the mobile device or changing the current state of an appliance (e.g. turning the TV off) sends an update to the *NetworkManager* and thus to every application that is currently running. The other applications receive an update event which allows them to update the state of the appliance. For example, when turning on the TV through *ProximityViewer*, all applications receive this update: *NetworkVisualizer* updates this information to present it; *TVClient* receives the update and turns off the TV; and finally, *LampAndClockClient* updates their model information but does not perform any action, as it does not have any functionality related to the TV.

5.3.3.2. Describing our System as a Distributed Model-View Controller (dMVC)

As shown in Figure 5.7 on page 99, our architecture follows a distributed Model-View-Controller (dMVC) design pattern (Boyle & Greenberg, 2005). This means that the centralized server (*NetworkManager*) holds a dictionary containing all

the *models* (known as the unified data model). In this case, a model is a data abstraction that describes each appliance or device (e.g. see Figure 5.9). Our `NetworkManager` acts as the *controller*, meaning that it is responsible for updating and modifying all of the models in our code. The *view*, corresponds to the controls' interfaces, as well as the behaviors on the appliances themselves (e.g. video playing on the TV, the radio showing the time) and receives these updates to perform the instructed functionality. Similarly, changing an individual interface element in the visual controls (part of the *view*) – such as tapping 'on' to turn the TV on – changes the model in that program and sends an update to the `NetworkManager`, which propagates updates to everything accordingly.

The `NetworkManager` receives updates about the mobile device position and orientation, as well as any appliances that may be tracked. This information can come from three sources: (1) the `ProximityToolkitTracker`, which feeds with information received from the Proximity Toolkit and the Vicon Motion Capture data; (2) the `LoggedDataTracker`, which takes pre-recorded information and replays it for debugging purposes³; or (3) the `ProximityViewer` client, which sends the compass and inclinometer information from the tablet. The `NetworkVisualizer` retrieves all information from the `NetworkManager` to present it for debugging purposes. This visualizer tool can be further extended to enable customization options.

Our models have been set up so that they contain attributes that correspond to the different types of content⁴ (see §3.2.2.2). This means it takes into account presence, state and controls. For example, Figure 5.9 shows the model of the lamp, where we can see attributes such as GUID (global unique identifier), name, type, icon, location, brightness and settings for turning of when people leave the room. The interface generator, in this case `ProximityViewer` can add further details

³ The data is stored into a text file containing the tablet's position (x, y, z) values and forward and up vectors with their corresponding (x, y, z) values

⁴ Our models also contain information about distance and orientation of the tablet for our programming convenience, but we would not expect this in actual deployment.


| LampModel | |
|-----------------|---|
| Guid | "DADE6DA4-834D-4C60-BC1A-23C9EDBB8A8A" |
| Name | "FloorLamp" |
| Type | Types.Movable |
| Location | (100, 1181, 1200) |
| IsOn | true |
| Brightness | 100 |
| TurnOffWhenAway | false |
| Icon |  |

Figure 5.9 Model of the Floor Lamp in our system

based on other information it may know, such as saving its own usage history for a device, or setting preferences. These types of extensions can also be done through the centralized server. As the server understands the configuration of the room and the appliances, it can have insight as to which appliances work

together and can inform ad-hoc connections that can be made between individual appliances. Our router shows an example of this by showing connected devices albeit being restricted as a mockup.

Another point worth mentioning is that we found that the Composite Pattern (Gamma, Helm, Johnson, & Vlissides, 1994) can serve as means to represent hierarchies or groups of appliances. In this case, our `RoomViewer` contains a reference to all the appliance models. Consequently our visual `RoomViewerController` has a model for each appliance and can operate them accordingly. In our case we implemented controls for the lamp and the television, but we can see how this can be extended to automatically generate interfaces for all connected appliances.

5.4 Limitations

Our system has several limitations that must be resolved before turning our proof-of-concept implementation into a real-world deployment. Our goal in this initial implementation was to make it effective enough to try out and evolve our design ideas. Even so, understanding these early limitations gives insight into issues that

a real system should address. From these, we propose potential solutions that can help in building a system that is more ready for deployment.

5.4.1 Scalability

Our first limitation is scalability – the number of appliances that our system is able to support. Our current ecology has seven items: the six appliances plus the room viewer. Some ecologies may incorporate a much larger number of devices and appliances. While our system can scale to accommodate more appliances, it raises the following question: *How can we deal with many interactive devices that are in very close proximity to each other?* Our HomeSpace positions appliances so that they are spread out around the room, or operating as a group. However, other rooms may have appliances closer together, such as a lamp next to – or even in front of – a thermostat. Our system *is* capable of showing more than one appliance control at a time depending on the positioning of the appliances. Yet simultaneously showing three or more controls for three or more appliances in close proximity can produce a confusing and overwhelming interface.

5.4.2 Configuration

The second limitation is configuration and rule setting. This concerns two different aspects. The first issue is how one can add or remove an appliance so that they can be recognized by the ecology. This also means that the system needs to understand whether an appliance needs to be tracked and how to can track it. In §3.1.1 we described how we can think of appliances according to their mobility and this brings some insight as to the extent they need to be tracked. For example, we can think of manual configurations in which we tell a system the location of an appliance. This works for static appliances (e.g., a wall-mounted thermostat). However, we still must account for small portable appliances such as a radio, which could be moved anytime and would require constant positioning information. Moreover, this manual configuration should also enable people to structure groups

of appliances. For example, if the system recognizes a television, a movie player and a home theater in close proximity, it may offer to group them as a hierarchy.

5.4.3 Rules of Engagement

Another point that we have not yet addressed is how to establish the rules of engagement. That is, determining the distance and orientation thresholds to enable presentation of information and controls of appliances as dictated by gradual engagement. As shown in this chapter, we established a range of two meters because we found that it worked for the size of the Home Space prototyping environment. Yet, this measurement may not be feasible or reasonable for a smaller or a larger space. Perhaps one way to deal with this is to establish measurements that depend on the dimensions of the room as a one-time configuration. Or, as part of a manual configuration process, users can establish exact distances and orientation thresholds to determine the interaction flow. While this rule set can be established for all appliances, perhaps some appliances which may be frequently unused may have more engagement restrictions, such as restricting the engagement area to a very small distance.

5.4.4 Architecture

The current architecture is extensible and works well for prototyping, although is impractical for field deployment. In particular, our architecture requires hand-crafted user interface controls for each appliance, each with its own look, feel and interaction capabilities. The controls are also crafted to create reasonable-looking continuous animated transitions that changed according to distance and orientation. Given the number of different appliances that could form an ecology, having the system contain a custom control for all possible appliances is unrealistic.

Another approach would be to create a standard representation, where each appliance can specify to the system both its capabilities and general UI attributes. For example, Nichols *et al.* proposed using XML as a way to encode user interfaces

for appliances (2002). The XML is sent to the device showing the appliance and is used to generate the interface. We could extend this idea further to incorporate more layout options and proxemic information, such as distances for continuous animations to start and end. Through this, the user's main application would be exclusively listening to events and generating interfaces as determined externally, e.g., as new appliances appear in the ecology. This approach may also mitigate another scalability issue, as the system does not need to know about any of the appliances *a priori*.

Another concern is whether we want a centralized server-based system, a completely distributed system, or perhaps a mix of the two. In a fully distributed system, appliances are responsible for directly communicating with the mobile devices. Consequently, appliances need computing power to send all the necessary information (e.g., such as the XML mentioned above), and the mobile device is responsible for sense-making of the information provided by the appliances, as well as having an understanding of the room, its configurations and its rules. Either or perhaps both will need to somehow track their relative locations from one another. This is perhaps ideal, as it means that devices can interact in a truly ad-hoc manner.

On the other hand, it is much easier to implement a centralized system that has a holistic view of the ecology, and that provides or coordinates communication and information exchange (such as tracking). Having a centralized system enables for easy configuration and to leverage high computational capabilities to even extend the functionality and intelligence of appliances. The server can be located in the actual space, where it controls only that space. Alternately, the server can be located in the cloud, where it could perhaps oversee multiple ecologies. Yet containing information on servers – particularly cloud-based servers – might pose some security risks for the people living in such homes⁵.

⁵ <http://www.popularmechanics.com/technology/how-to/gadgets/i-automated-my-apartment-and-it-kind-of-creeped-me-out-16708938>

5.4.5 Technical

The next limitation of our prototype is the technical setup. Currently we require a sophisticated motion capturing system to track the tablet and other moveable entities. This is clearly impractical. It is overly costly, physically constrained to a room-sized space, and requires constant re-configuration. The many cameras are also intrusive. Newer systems, such as the Kinect, are far more practical but are still limited in how they can identify devices and their orientation, and in their inability to handle occlusion. We can imagine how in the near future we can leverage the mobile devices themselves (as done in Google's Project Tango⁶) to understand their own position and orientation in space via its own ability to track 3d motion and depth sensing.

Networking is another limitation. Our server-based approach relies on a TCP/IP connection over an established local network (e.g., wireless, Ethernet). That is, it relies on local infrastructure to supply the network. In contrast, a distributed system can use on-board networking (such as Bluetooth or Zigbee) to manage communication between the tablet and the appliances that surround it.

5.4.6 Evaluation

Finally, this thesis encompasses two design and implementation cycles with a multitude of variations within each. Our decisions and subsequent design choices were done through self-reflection (i.e., our own experiences critiquing and trying out our system), as well as from feedback and discussion with the many people who saw various forms of our system in live demonstrations. However, this thesis does not tell us whether our system can be realistically applied to home environments and what kind of experience it creates for people. We believe our prototype is not yet ready for a field deployment as we currently do not have the technological means to install this in people's homes. It would require us to set up a sophisticated tracking system, establish the necessary network protocols and hack individual

⁶ <https://www.google.com/atap/projecttango/>

appliances to work with our system. It also means we would have to customize individual interfaces for each room.

Instead of ‘on the field’ validation, we could have used various ‘lab-based’ methods to validate our design as is typically done in the HCI research community. For example, we could have an in-lab comparative study that contrasts our gradual engagement approach to touching, pointing and scanning. However, this approach may not directly answer questions pertaining the overall user experience. We could, perhaps, test individual parts of our prototype with proximity controls turned against a version of it that just required people to select and activate those appliances manually (i.e., locking the screen, selecting from the overview, and then moving the slider). Yet this would be premature, as usability issues that are not central to our thesis may have considerable effect on the result. A better approach would be to have people try out the system in the lab while thinking-aloud, perhaps followed by retrospective interviews where we discuss their reactions to it. This could reveal usability issues, and also their perception of how they view gradual engagement with appliances. Even so, their experiences would be heavily colored by the particular appliances we chose, and how we explain the scenarios of use. The novelty of our approach compared to more traditional methods such as pointing may also confound the results of our observations.

Clearly, a formal evaluation needs to be done, but it is beyond the scope of this thesis – which focuses on design and implementation of a proof-of-concept prototype. We believe a worthwhile evaluation will require much additional thought and care, and will help guide further evolution of the design concepts presented here.

Chapter 6. Conclusions and Future Work

This chapter serves as closing remarks to this thesis, which explored how **we can leverage proxemics as a socially established protocol to create seamless interaction with appliances**. This begins with the problem definition from Chapter 1 pertaining the difficulties of knowing *which* devices are interactive, *how* they can be selected, *what* information they contain and *how* one can reveal pertinent controls. The problems are followed by the goals and then a reflection as to how the thesis contributions addressed these problems (§6.1). I then discuss potential future work that goes beyond this particular project and towards other visions that can be directly applied to Human-Computer Interaction (§6.2). I finish with closing remarks reflecting on the work presented and what I believe to be the impact of this research (§6.3).

6.1 Revisiting Thesis Problems and Solutions

In Chapter 1, I introduced the problems that resulted from introducing an increasing number of appliances within a ubicomp ecology, and its effect on *how a person can control and interact with all these appliances*. I explained how, to mitigate these problems, I would adapt the notion of gradual engagement to satisfy interactions with ubicomp ecologies in a broader sense and enable control of appliances within a room. This section reiterates the problems, the goal that mitigates the problems and our proposed solutions.

Chapter 6 • Conclusions and Future Work

| | |
|----------------------------|---|
| Problem #1 | It is difficult for people to know which appliances possess interactive capabilities. |
| Problem Description | When a person walks into a room, there is no way for them to know (1) the location of different appliances; and (2) whether an appliance can be controlled. |
| Goal | Discover interactive appliances within a room. |
| Proposed Solution | Our gradual engagement of controls introduced in Chapter 3 §3.2.2 depicts a mechanism to find interactive appliances. This relies on two components. The first way is by orienting a device towards an interactive appliance to reveal content that varies according to the physical distance between the device and the appliance. The second mechanism is through the overview techniques introduced in §3.2.3.1, extending prior work on visualizations for spatial references. These spatial references provide individuals with awareness of the relative position of the interactive appliances, which can be used to find the physical counterpart of the appliance. |
| Problem #2 | It is difficult to select an individual appliance within an ecology of devices. |
| Problem Description | As the number of devices in the ecology increases, it becomes increasingly difficult to select an individual appliance in order to control it. |
| Goal | Select individual appliances. |
| Proposed Solution | Our contextualization of gradual engagement establishes how we can leverage orientation as a way to engage with an interactive appliance (Chapter 3 |

Chapter 6 • Conclusions and Future Work

§3.2.2 and §3.2.3.2). Chapter 4 shows how we apply this notion while §5.2 explains how we make use of distance and orientation to position controls on the tablet display. Moreover, we explained how our approach leverages manual override and thus enables integration of other interaction techniques such as scanning (i.e. showing a list of devices that can be selected), touching and appliance, and pointing towards it (§4.2.3).

| | |
|----------------------------|--|
| Problem #3 | It is difficult to provide a large amount of information about an appliance. |
| Problem Description | Appliances typically contain a lot of information, such as indication of being on or off, the current task being performed, or how much battery power remains. While typically a remote control does not enable users to see all of these items, this information should be accessible when warranted. |
| Goal | View information about appliances. |
| Proposed Solution | In §3.2.2 and §3.2.3.3 we showed how we can use gradual engagement of controls to present information continuously as a function of distance. Through this we have been able to show the contents of the interface so that the transition as one gets closer to an appliance enables individuals to see more information about an appliance and thus leverage usability and flexibility. Many of our scenarios in Chapter 4 illustrate this as well. |

| | |
|----------------------------|---|
| Problem #4 | It is difficult to provide pertinent controls for appliances |
| Problem Description | This problem is similar to revealing information about an appliance, however it is concerned specifically with how and when controls should be presented and focuses on enabling a user to act on an appliance as opposed to simply view their content. These controls should be usable and provide flexibility and leverage the mobile devices' limited display sizes. |
| Goal | Control appliances. |
| Proposed Solution | Our gradual engagement concept can be applied to revealing controls similarly to how it is applied to revealing information in the solution to problem 3. Furthermore, our application of the radio alarm clock (Chapter 4 §4.2.4) demonstrates how we can use the space around an appliance to distribute controls. Similarly our grouping techniques (hierarchies) enable aggregation of controls and thus simplify the interfaces presented on screen. |

6.2 Contributions

With this thesis, I have used the concept of proxemics to inform the design of seamless interaction with appliances in a ubicomp ecology through the use of socially acceptable protocols that can be understood by people (proxemic interaction). As a result, I have made the following contributions to the field of Human–Computer Interaction.

6.2.1 Major Contributions

1. **A framework describing proxemic-aware controls.** Our design directly applies Marquardt et al.'s Gradual Engagement design pattern (2012) to the context of appliances. Through this process, I have shown how we can pertinently reveal content about appliances in a room. I have illustrated interfaces that can change their level of complexity dynamically and that can quickly transition from controlling one appliance to another.
2. **A prototype operationalizing gradual engagement of controls.** The proxemic-aware controls prototype implementation demonstrates how we can think beyond the design vision and how we can directly apply the concepts of gradual engagement as a proof-of-concept. Furthermore, I was able to iterate and refine the concepts to show how a system like this can be applied in the real world once the technology is readily available.

6.2.2 Minor Contributions

1. An application of a spatial user interface that demonstrates continuous flow of information as a function of distance.
2. A demonstration of proxemic interaction that makes use of a larger ecology of devices: one tablet and six appliances.
3. A spatial application of Shneiderman's information visualization mantra showing overview first and details on demand.
4. A means to integrate known interaction techniques with appliances. Gradual engagement of controls enables users to transition between scanning, touching and pointing.

6.3 Avenues for Future Work

Chapter 5 (§5.4) detailed the limitations to this work. Overcoming these limitations could lead to further extensions specific to this particular context of interaction

with appliances with traditional mobile devices. However, this section aims to describe potential avenues for future work as a vision that stems from the general concepts presented throughout this thesis.

6.3.1 Architecture that Enables Ad-Hoc Proxemic Information

As mentioned in the previous §5.4, one limitation of our architecture is that the mobile device is responsible for storing all interface controls for each interactive appliance. One possible extension is to create a system and architecture that is able to transfer information directly from the server (or the cloud) to the mobile device. As mentioned earlier, one way to do this is to use previous work in XML encoding to describe interfaces (Nichols et al., 2002), in which we can add encoding for proxemic distances and adjustable parameters such as continuous versus discrete engagement. Furthermore, it would be interesting to explore how we can create authoring tools that enable designers to create these controls with ease and be able to associate user interface controls (e.g. buttons and other graphical elements) to appliance input. This could mean the creation of a protocol that transfers encoded information and that has tools that enable access and customization to less technical users.

6.3.2 Exploring and Comparing Different Form Factors

This thesis has largely focused on the use of a mobile device such as a phone or tablet to interact with appliances. However, certain variations in form factor may require further extensions to the gradual engagement pattern. For instance, mobile devices such as augmented reality glasses or smart watches may require different presentation techniques, or ways to think about our device-to-appliance interaction, as smart watches are attached to a user's wrist. Similarly, we can consider how the paradigm changes when information can be projected atop or around the appliance itself, such as through portable pico-projectors or by projecting all information in the room. Xiao et al. made use of a spherical mirror

to extend cursors beyond a single display by allowing them to transition into physical space through projections on the wall (2011). However, some of their unpublished work in progress shows this scenario further extended to augment a room, such as projecting information on top of a printer. We can thus think about how a projection-augmented room provides new opportunities for presenting content and new interaction challenges. The contribution would be two-fold: first, we would explore and discover new ways of interacting with ecologies of appliances through unconventional modalities (e.g. glasses, projection, smart watches); and second, we could further generalize our gradual engagement of control pattern to account for different design constraints resulting from the different types of devices.

6.3.3 Applying Mobile Gradual Engagement to the Physical World

One of the main points in this thesis was the inclusion of everyday appliances and devices into ubicomp ecologies. This addition provided us with new interaction challenges and possibilities and enabled us to apply gradual engagement with a mobile device as the center of interaction. One potential avenue would be to incorporate these concepts into other environments and view *in-situ* anchored information in the physical world, such as in public spaces. For example, we can apply the same ideas of gradual engagement as a way to explore a museum exhibit. Once a person enters a room they can see the different artifacts, and physically approaching them enables them to see more information and media, such as videos or visualizations describing the installations. Hinrichs (2013) showed that in public spaces, people can engage in information exploration activities pertaining to a museum or gallery exhibition on large displays, where their interactions with the interactive visuals can lead to new discoveries. Similarly, our application of gradual engagement opens up opportunities to be able to explore information and digital content with a mobile device. It can encourage museum visitors to learn

more about the exhibition beyond a standard audio guide. For some exhibits, a visitor may even be able to interact with it or affect an installation's behavior.

6.3.4 Prototyping New Deployable Tracking Technologies

One major limitation to making our system accessible to a typical living room is the need for sophisticated tracking technologies. The Vicon system we use is not practical nor desirable for real world deployment. A feasible form of tracking technology is clearly essential, as we need to understand the positioning information of an object in space. One possible approach is to make use of computer vision techniques to understand position and orientation. Dearman et al. made use of image stitching on different mobile cameras to understand location and orientation data (2012). Similarly, we can explore ways to leverage feature recognition to understand position and orientation of different appliances. Another way to track devices can be through high-frequency sounds that are inaudible to the human ear but easily recognizable by a mobile device's microphone. We can think of augmenting appliances with speakers that play a unique acoustic barcode as a combination of high frequency sounds with a starting point and an ending point. The mobile device then recognizes the sound and uses the cloud or the server to retrieve the correct interface. Furthermore, the volume level can be used as a way to understand distance. Another approach is to position three speakers in different corners of the room each playing a different frequency and using that to triangulate a location, similar to Marquardt et al.'s triangulation in their GroupTogether system (2012). However, these ideas are speculative at best. What is clear is that some kind of tracking technology – one that is cheap, easy to deploy and ideally required no fixed infrastructure – is needed.

6.4 Closing Remarks

As appliances become capable of communicating with each other and with other devices, they provide more opportunities for interacting with them and controlling them. My hope is that this thesis can serve as a starting point to open up new

possibilities for appliance interaction and to encourage novel ways of creating user interfaces that go beyond conventional menu navigation approaches, such as current standard user interfaces. With our gradual engagement of controls and our system, we have informed the design of remote controls for increasingly large ecologies of devices. We have shown that these interactions do not have to be restrictive of traditional interaction approaches such as pointing at an appliance to control it. Instead, these interactions can coexist and enable a user to choose the most pertinent action depending on the task at hand. Furthermore, the mobile application of gradual engagement holds great potential for other application areas beyond controlling appliances and can bring our interactions to our social channel to create a seamless user experience, where we understand the established protocols without the need to disconnect our on-device experience with our outside-world experience.

References

- Ballendat, T., Marquardt, N., & Greenberg, S. (2010). Proxemic interaction: designing for a proximity and orientation-aware environment. In *ACM International Conference on Interactive Tabletops and Surfaces* (pp. 121–130). New York, NY, USA: ACM. doi:10.1145/1936652.1936676
- Bederson, B. B., & Hollan, J. D. (1994). Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology* (pp. 17–26). New York, NY, USA: ACM. doi:10.1145/192426.192435
- Beigl, M. (1999). Point & Click-Interaction in Smart Environments. In H.-W. Gellersen (Ed.), *Handheld and Ubiquitous Computing* (pp. 311–313). Springer Berlin Heidelberg. Retrieved from http://link.springer.com/chapter/10.1007/3-540-48157-5_31
- Boring, S., Baur, D., Butz, A., Gustafson, S., & Baudisch, P. (2010). Touch projector. In *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10* (p. 2287). Atlanta, Georgia, USA. doi:10.1145/1753326.1753671
- Boyle, M., & Greenberg, S. (2005). Rapidly Prototyping Multimedia Groupware. In *DMS* (pp. 178–183).

References

- Buxton, B. (2010). *Sketching User Experiences: Getting the Design Right and the Right Design: Getting the Design Right and the Right Design*. Morgan Kaufmann.
- Chen, Y.-H., Zhang, B., Tuna, C., Li, Y., Lee, E. A., & Hartmann, B. (2013). A Context Menu for the Real World: Controlling Physical Appliances Through Head-Worn Infrared Targeting.
- Dearman, D., Guy, R., & Truong, K. (2012). Determining the Orientation of Proximate Mobile Devices Using Their Back Facing Camera. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 2231–2234). New York, NY, USA: ACM. doi:10.1145/2207676.2208377
- Dourish, P. (2004). *Where the Action is: The Foundations of Embodied Interaction*. MIT Press.
- Dow, S. P., Glassco, A., Kass, J., Schwarz, M., Schwartz, D. L., & Klemmer, S. R. (2010). Parallel Prototyping Leads to Better Design Results, More Divergence, and Increased Self-efficacy. *ACM Trans. Comput.-Hum. Interact.*, 17(4), 18:1–18:24. doi:10.1145/1879831.1879836
- Gaines, B. R. (1991). Modeling and forecasting the information sciences. *Information Sciences*, 57–58, 3–22. doi:10.1016/0020-0255(91)90066-4
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Pearson Education.
- Gellersen, H., Fischer, C., Guinard, D., Gostner, R., Kortuem, G., Kray, C., ... Streng, S. (2009). Supporting device discovery and spontaneous interaction

References

- with spatial references. *Personal Ubiquitous Comput.*, 13(4), 255–264.
doi:10.1007/s00779-008-0206-3
- Goodson, L. E., McIntyre, W. R., & Rumbolt, R. B. (1987). *Universal remote control unit with model identification capability*. Google Patents.
- Greenberg, S., & Fitchett, C. (2001). Phidgets: easy development of physical interfaces through physical widgets. In *Proceedings of the 14th annual ACM symposium on User interface software and technology* (pp. 209–218). Orlando, Florida: ACM. doi:10.1145/502348.502388
- Greenberg, S., & Kuzuoka, H. (1999). Using digital but physical surrogates to mediate awareness, communication and privacy in media spaces. *Personal Technologies*, 3(4), 182–198. doi:10.1007/BF01540552
- Greenberg, S., Marquardt, N., Ballendat, T., Diaz-Marino, R., & Wang, M. (2011). Proxemic interactions: the new ubicomp? *Interactions*, 18(1), 42–50. doi:10.1145/1897239.1897250
- Hall, E. T. (1969). *The hidden dimension*. Anchor Books New York.
- Hausen, D., Boring, S., Lueling, C., Rodestock, S., & Butz, A. (2012). StaTube: Facilitating State Management in Instant Messaging Systems. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction* (pp. 283–290). New York, NY, USA: ACM. doi:10.1145/2148131.2148191
- He, H. A. (2010, April). *One Size Does Not Fit All: Extending the Transtheoretical Model to Energy Feedback Technology Design*. Department of Computer Science, University of Calgary, Calgary, Alberta, Canada.

References

- Hinckley, K. (2003). Synchronous Gestures for Multiple Persons and Computers. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology* (pp. 149–158). New York, NY, USA: ACM. doi:10.1145/964696.964713
- Hinrichs, U. (2013). Open-Ended Explorations in Exhibition Spaces: A Case for Information Visualization and Large Direct-Touch Displays.
- Ishii, H., Wisneski, C., Brave, S., Dahley, A., Gorbet, M., Ullmer, B., & Yarin, P. (1998). ambientROOM: Integrating Ambient Media with Architectural Space. In *CHI 98 Conference Summary on Human Factors in Computing Systems* (pp. 173–174). New York, NY, USA: ACM. doi:10.1145/286498.286652
- Kim, D., Hilliges, O., Izadi, S., Butler, A. D., Chen, J., Oikonomidis, I., & Olivier, P. (2012). Digits: freehand 3D interactions anywhere using a wrist-worn gloveless sensor. In *Proceedings of the 25th annual ACM symposium on User interface software and technology* (pp. 167–176). New York, NY, USA: ACM. doi:10.1145/2380116.2380139
- Kohtake, N., Rekimoto, J., & Anzai, Y. (2001). InfoPoint: A Device that Provides a Uniform User Interface to Allow Appliances to Work Together over a Network. *Personal and Ubiquitous Computing*, 5(4), 264–274. doi:10.1007/s007790170005
- Kopetz, H. (2011). Internet of Things. In *Real-Time Systems* (pp. 307–323). Springer US. Retrieved from http://link.springer.com/chapter/10.1007/978-1-4419-8237-7_13

References

- Ledo, D., & Greenberg, S. (2013). Mobile Proxemic Awareness and Control: Exploring the Design Space for Interaction with a Single Appliance. In *Proc. ACM CHI 2013 Video Program (Juried) - CHI'13* (p. 1 page abstract plus video). Paris, France.
- Lehikoinen, J., & Suomela, R. (2002). Accessing Context in Wearable Computers. *Personal Ubiquitous Comput.*, 6(1), 64–74. doi:10.1007/s007790200006
- Lidwell, W., Holden, K., & Butler, J. (2003). *Universal Principles of Design: 100 Ways to Enhance Usability, Influence Perception, Increase Appeal, Make Better Design Decisions, and Teach Through Design*. Rockport.
- Luplow, W. C., & Taylor, J. I. (2012). Channel Surfing Redux: A Brief History of the TV Remote Control and a Tribute to Its Coinventors. *Consumer Electronics Magazine, IEEE*, 1(4), 24–29.
- Marincic, A. (1998). A Century of Remote Control. *Microwave Review*.
- Marquardt, N. (2013, July). *Proxemic Interactions in Ubiquitous Computing Ecologies*. Department of Computer Science, University of Calgary, Calgary, Alberta, Canada.
- Marquardt, N., Ballendat, T., Boring, S., Greenberg, S., & Hinckley, K. (2012). Gradual Engagement between Digital Devices as a Function of Proximity: From Awareness to Progressive Reveal to Information Transfer. In *Proceedings of Interactive Tabletops a& Surfaces - ACM ITS* (p. 10 pages). Boston, USA: ACM Press.
- Marquardt, N., Diaz-Marino, R., Boring, S., & Greenberg, S. (2011). The proximity toolkit: prototyping proxemic interactions in ubiquitous computing

References

- ecologies. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (pp. 315–326). New York, NY, USA: ACM. doi:10.1145/2047196.2047238
- Marquardt, N., Hinckley, K., & Greenberg, S. (2012). Cross-device Interaction via Micro-mobility and F-formations. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (pp. 13–22). New York, NY, USA: ACM. doi:10.1145/2380116.2380121
- Mulloni, A., Dünser, A., & Schmalstieg, D. (2010). Zooming interfaces for augmented reality browsers. In *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services* (pp. 161–170). New York, NY, USA: ACM. doi:10.1145/1851600.1851629
- Myers, B. (2002). Mobile Devices for Control. In F. Paternò (Ed.), *Human Computer Interaction with Mobile Devices* (Vol. 2411, pp. 569–579). Springer Berlin / Heidelberg. Retrieved from <http://www.springerlink.com/content/9dmkxjcdveyedn2m/abstract/>
- Myers, B. A., Peck, C. H., Nichols, J., Kong, D., & Miller, R. (2001). Interacting at a Distance Using Semantic Snarfing. In G. D. Abowd, B. Brumitt, & S. Shafer (Eds.), *Ubicomp 2001: Ubiquitous Computing* (Vol. 2201, pp. 305–314). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from <http://www.springerlink.com/content/4em3f105x1uqb470/>
- Nichols, J., Myers, B. A., Higgins, M., Hughes, J., Harris, T. K., Rosenfeld, R., & Pignol, M. (2002). Generating Remote Control Interfaces for Complex

References

- Appliances. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology* (pp. 161–170). New York, NY, USA: ACM. doi:10.1145/571985.572008
- Nichols, J., Rothrock, B., Chau, D. H., & Myers, B. A. (2006). Huddle: automatically generating interfaces for systems of multiple connected appliances. In *Proceedings of the 19th annual ACM symposium on User interface software and technology* (pp. 279–288). New York, NY, USA: ACM. doi:10.1145/1166253.1166298
- Owen, C. L. (1998). Design research: building the knowledge base. *Design Studies*, 19(1), 9–20.
- Plaisant, C., & Shneiderman, B. (1992). Scheduling home control devices: design issues and usability evaluation of four touchscreen interfaces. *International Journal of Man-Machine Studies*, 36(3), 375–393. doi:10.1016/0020-7373(92)90040-R
- Remote Control Anarchy (Jakob Nielsen's Alertbox). (2004). Retrieved October 29, 2012, from <http://www.useit.com/alertbox/20040607.html>
- Rogers, Y. (2004). New theoretical approaches for human-computer interaction. *Annual Review of Information Science and Technology*, 38(1), 87–143.
- Rukzio, E., Leichtenstern, K., Callaghan, V., Holleis, P., Schmidt, A., & Chin, J. (2006). An Experimental Comparison of Physical Mobile Interaction Techniques: Touching, Pointing and Scanning. In P. Dourish & A. Friday (Eds.), *UbiComp 2006: Ubiquitous Computing* (Vol. 4206, pp. 87–104).

- Springer Berlin / Heidelberg. Retrieved from <http://www.springerlink.com/content/h6jut558g604x083/abstract/>
- Schmidt, D., Molyneaux, D., & Cao, X. (2012). PICOntrol: using a handheld projector for direct control of physical devices through visible light. In *Proceedings of the 25th annual ACM symposium on User interface software and technology* (pp. 379–388). New York, NY, USA: ACM. doi:10.1145/2380116.2380166
- Seifried, T., Haller, M., Scott, S. D., Perteneder, F., Rendl, C., Sakamoto, D., & Inami, M. (2009). CRISTAL: a collaborative home media and device controller based on a multi-touch display. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (pp. 33–40). New York, NY, USA: ACM. doi:10.1145/1731903.1731911
- Shneiderman, B. (1996). The eyes have it: a task by data type taxonomy for information visualizations. In , *IEEE Symposium on Visual Languages, 1996. Proceedings* (pp. 336–343). doi:10.1109/VL.1996.545307
- Swindells, C., Inkpen, K. M., Dill, J. C., & Tory, M. (2002). That One There! Pointing to Establish Device Identity. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology* (pp. 151–160). New York, NY, USA: ACM. doi:10.1145/571985.572007
- Tani, M., Yamaashi, K., Tanikoshi, K., Futakawa, M., & Tanifuji, S. (1992). Object-oriented video: interaction with real-world objects through live video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing*

References

- Systems* (pp. 593–598). New York, NY, USA: ACM.
doi:10.1145/142750.143011
- Välkkynen, P., & Tuomisto, T. (2005). Physical Browsing Research. *PERMID, 2005*, 35–38.
- Vogel, D., & Balakrishnan, R. (2004). Interactive Public Ambient Displays: Transitioning from Implicit to Explicit, Public to Personal, Interaction with Multiple Users. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology* (pp. 137–146). New York, NY, USA: ACM. doi:10.1145/1029632.1029656
- Want, R., Fishkin, K. P., Gujar, A., & Harrison, B. L. (1999). Bridging physical and virtual worlds with electronic tags. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (pp. 370–377). New York, NY, USA: ACM. doi:10.1145/302979.303111
- Weiser, M. (1999). The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3(3), 3–11. doi:10.1145/329124.329126
- Wiberg, M., & Stolterman, E. (2014). What Makes a Prototype Novel?: A Knowledge Contribution Concern for Interaction Design Research. In *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational* (pp. 531–540). New York, NY, USA: ACM. doi:10.1145/2639189.2639487
- Xiao, R., Nacenta, M. A., Mandryk, R. L., Cockburn, A., & Gutwin, C. (2011). Ubiquitous Cursor: A Comparison of Direct and Indirect Pointing Feedback in Multi-display Environments. In *Proceedings of Graphics Interface 2011*

References

- (pp. 135–142). School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada: Canadian Human-Computer Communications Society. Retrieved from <http://dl.acm.org/citation.cfm?id=1992917.1992939>
- Yoon, H., Kim, H., & Woo, W. (2007). Personal Companion: Personalized User Interface for U-Service Discovery, Selection and Interaction. In C. Stephanidis (Ed.), *Universal Access in Human-Computer Interaction. Ambient Interaction* (Vol. 4555, pp. 1052–1061). Springer Berlin / Heidelberg. Retrieved from <http://www.springerlink.com/content/fg815857t2q13141/abstract/>
- Zimmerman, J., Forlizzi, J., & Evenson, S. (2007). Research Through Design As a Method for Interaction Design Research in HCI. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 493–502). New York, NY, USA: ACM. doi:10.1145/1240624.1240704

Appendix – Methodology and Process

This appendix (summarized in Figure A.1) describes the methodology and process that was used in this research. This chapter *retroactively* describes the methodology as *research through design* and identifies and explains the four approaches used concurrently throughout the process: design guidelines, concept design, generic design and proof-of-concept design. I discuss each approach, the results and the lessons learned, and then the interplay between the approaches. It is important to note that the insights collected throughout each of the stages were the results of self-reflection, as well as critical discussions during live demonstrations in our laboratory to over 150 people from industry (e.g. SMART Technologies, TR Tech), public organizations (Alberta Health Services, Alberta Innovates Technology Futures) and visiting academics from around the world (Canada, United States, France, Denmark, Germany, United Kingdom, etc.).

This thesis utilizes a *research through design* methodology, the focus of which is to produce knowledge guiding towards the *right design*, as opposed to a *commercially successful product* (Zimmerman et al., 2007). The rationale behind adopting this research methodology is that it enables going beyond seeking an individual solution to a problem, rather it leads to an understanding of the larger conceptual space from which multiple solutions can be drawn. Furthermore, this research through design methodology provides a means to articulate the process undertaken throughout the different iterations of our design (shown in Chapter 4). From a simplified point of view, the design process can be seen as a combination of **top-down** (using knowledge to inform implementation) and **bottom-up** (creating prototypes to guide exploration and provide reflection) approaches.

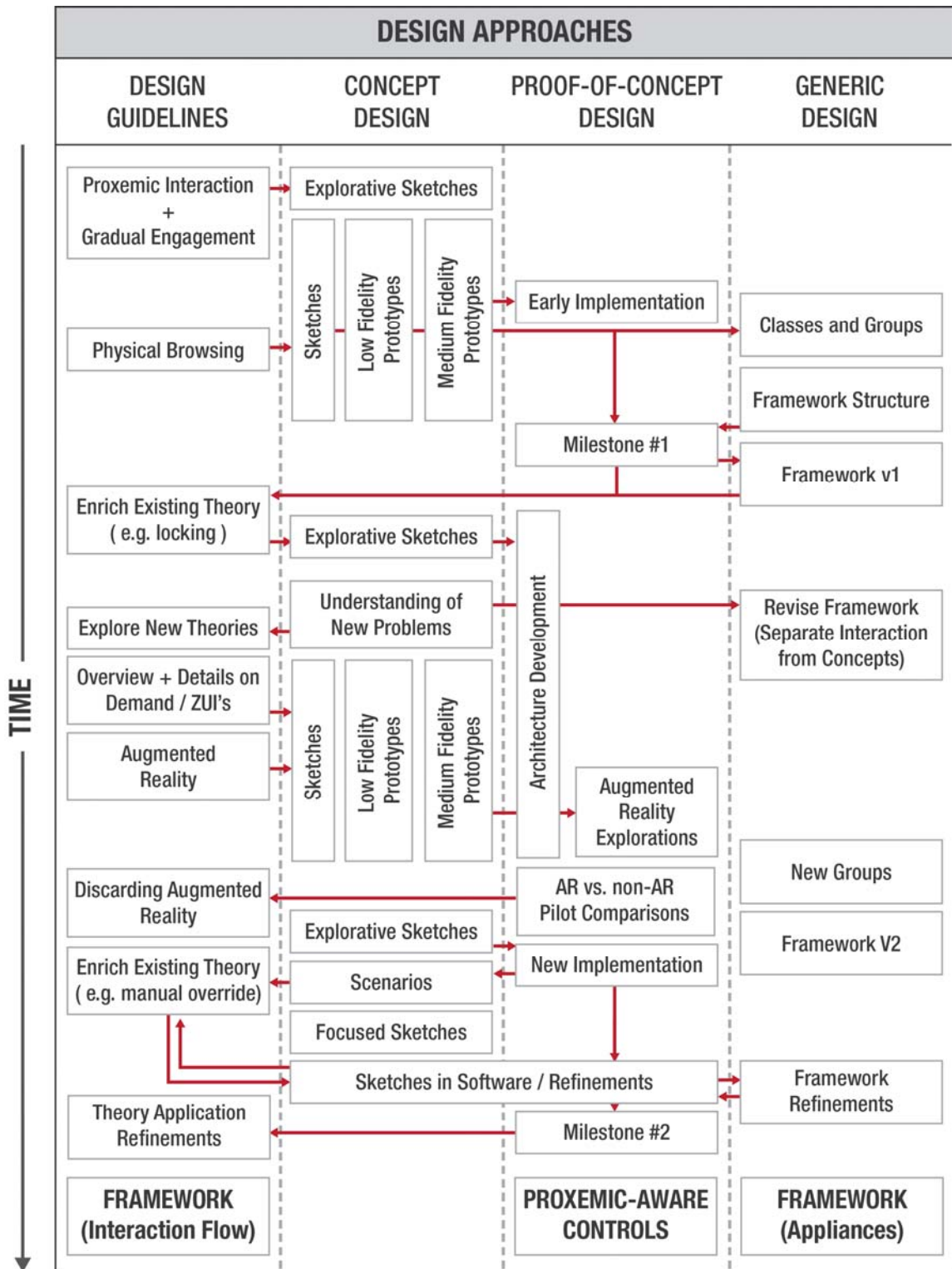


Figure A.1 Summary of the design process undertaken in this thesis from the lens of research through design methodology.

| | | OVERALL PURPOSE | WAYS OF WORKING IDEAS | MAIN ACTIVITY | OUTCOME OR GOAL | RELATION TO DESIGN SPACE |
|-------------------|-------------------------|-----------------|------------------------------|-----------------------|-----------------------|--|
| DESIGN APPROACHES | PROOF-OF-CONCEPT DESIGN | Verify design | Concrete Properties | Build and test | Plausible design | Demonstrate the possibility of the realization of a design space |
| | DESIGN GUIDELINES | Guide Design | Abstract Properties | Specify and formulate | Complete design | Define a design within a design space |
| | CONCEPT DESIGN | Position Design | Concrete Points of Reference | Sketch and model | New design | Points to a possible design space |
| | GENERIC DESIGN | Describe Design | Abstract Points of Reference | Group and Describe | Characteristic design | Defines a design space |

Figure A.2 Design approaches as depicted by Wiberg and Stolterman (2014).

This research through design model was proposed by Zimmerman et al. and emphasizes the contribution of novel integrations of theory and technology, in which the interaction design is applied in a new context, in this case remote controls (ibid). Towards maintaining a scientific approach, research through design can be evaluated in terms of *process* (to enable reproducibility), *invention* (a significant invention that addresses a problem), *relevance* (as opposed to validity, the focus of research through design is relevance, and help the community in considering the proposed work), and *extensibility* (how others can build on the resulting outcome) (ibid). The process presented in this chapter also includes lessons learned and reflections that have guided our design process until reaching the final prototype.

Akin to Wiberg and Stolterman's approaches to design (2014), the work in this thesis follows: **design guidelines** through the use of theory within HCI to drive the concept exploration (§A.1); **concept design** through sketches and models which move towards a possible design space (§A.2); **generic design** by compiling

these explorations into a framework that describes the design space (§A.3); and ***proof-of-concept design***, in which we illustrate some of the concepts through a high-fidelity prototype that demonstrates the possible realization of the proposed design space (§A.4). Figure A.2 shows a table illustrating the approaches, their targets and how they serve to describe a design space. This chapter then describes the process undertaken and how it corresponds to the methodology (§A.5), as well as the lessons learned (§A.6). It is important to stress that this lens of looking at our work through different design approaches has been taken *retroactively*, as these explorations are fairly recent in the research community. The following subsections elaborate on these approaches in the context of this thesis, and shows both how these different approaches are complementary to each other throughout every stage and where the actions from different steps of the process interweave themselves with other components in each design approach. Figure A.1 shows this interwoven process with the different stages of the research process. The arrows in the figure show the influence from one approach to the other.

A.1 Design Guidelines through HCI Theory

A key component of the design process of this thesis is that it is theory-driven. In this case, the theory serves to drive and focus the design exploration in the conceptual stage. As explained by Yvonne Rogers (2004), HCI research tends to have a strong separation between theory and applications, yet theoretical explorations (often in the form of frameworks) can enable designers and researchers to “better articulate and theoretically ground the challenges facing them today” (pp. 87). Of particular interest are *formative* (establishing vocabulary and concepts to discuss designs) and *generative* (providing constructs to inform design) theories (ibid). Rogers also suggests that synthesizing concepts from more than one theory can lead to contextualized frameworks that can be more robust and provide a broader set of concepts from which to think about design (ibid). From an information sciences perspective, the use of existing theories as depicted in this thesis encourages *replication* of existing ideas with small variations to

increase experience in order to reach *empiricism* over time (gaining a better understanding of rules that have been found useful) (Gaines, 1991). Under these criteria, having a theory-driven design allows for grounding ideas on already established concepts while still providing freedom to generate multiple designs, which are the foundations for design thinking (Zimmerman et al., 2007). The choice of theories to drive the design in this thesis is not arbitrary, rather *they are a set of tools that can help better target and surround the problems at hand* depicted in Chapter 1.

A.2 Concept Design

Throughout the entire design process, concept design took place in the form of sketches and implementation, the former serving as a constant tool for thinking and communicating concepts. While the implementation serves as the proof-of-concept design, smaller individual implementations—along with the different iterations and refinements explored—serve to examine different design variations thus becoming software sketches. These explorations follow some of the ideas portrayed in Buxton's *Sketching User Experiences* (2010). First, free-range explorations through sketching allow for exploring breadth, that is, *getting the right design(s)* that does explore multiple solutions, as opposed to *getting a single design right*. The sketches offer glimpses of solutions that can be subjectively appropriate or ill fitting. Second, as the design process goes along, an expansion and reduction of ideas takes place and thus leads to a more focused design, which Buxton describes as the design funnel (ibid). Finally, sketching enables for rapid parallel prototyping, which leads to better design results and more breadth of solutions (Dow et al., 2010). Although we based our concept design on proxemic interaction theory (discussed in §A1), we were still able to do considerable expansion while still having constraints.

A.3 Generic Design

As it has been shown throughout this thesis, the use of HCI theory provides design guidelines that ground and structure the design process, and the concept design expands upon many design alternatives that describe the space. Wiberg and Stolterman (2014) suggest four steps for design conceptualization: (1) *identifying* existing groups, classes, modes and genres in HCI; (2) *relating* existing designs to these groups and classes; (3) *formulating* similarities and differences between the different groups and provide definitions for each class; and (4) *designing and implementing* different designs that fit into the different classes. While these steps are shown sequentially, the work in this thesis constantly transitioned between each stage in a non-sequential order, which highlights the interplay between the different design approaches (i.e. seeing how they inform each other) taken in parallel (Figure A.1, Page 143). Ultimately, the outcome of the generic design is considered to be the knowledge contribution of this thesis, in this case, the framework outlined in Chapter 3.

A.4 Proof-Of-Concept Design

Wiberg and Stolterman (2014) describe proof-of-concept design as a means to verify a design at hand by showing how particular components of a design space can be realized. In the same way, the different implementation milestones shown in this thesis are the illustration of the design process up to that point (discussed in chapters 4 and 5 and later in this section). That is, the proof-of-concept was directly guided by the theoretical guidelines (§A1), and the concepts conveyed were selected from the conceptual sketches in a way that they illustrated different aspects of the framework, our generic design. Having a real-world application and being able to test it ourselves and demonstrate it to people led to new insights, as the reflection led us to realize what worked and did not work in each case, which would take us back to search for new theories to inform our designs, as well as initiate a new expansion process in our sketches thus leading to new framework

components. In a similar fashion, the lessons drawn from the proof-of-concept can refine the knowledge provided by the theory.

A.5 Process and Lessons Learned

Now that the methodological approach has been exposed, this section explains the design process in a somewhat linear manner. Referring back to Figure A.1 on page 143, this process had a lot of the different approaches done in parallel, in which the results of one stage would serve to inform other components (indicated by an arrow). The next subsections will explain some of the highlights of the design process, which spanned over two years.

A.5.1 Defining the Initial Problem Statement

The initial motivation and problem statement for this research stemmed from everyday appliances having very limited input and output. In contrast, mobile devices provide with rich interaction mechanisms through their high resolution screens and multi-touch capabilities. This was the starting point for this work, and it progressed as new problems were discovered later in the process.

A.5.2 First Theoretical Enquiry – Proxemic Interaction and Gradual Engagement

Our initial approach was to use proxemic interaction and gradual engagement, as we found it entailed a means reveal information and controls for appliances to make up for their limited input and output capabilities, while accommodating for the usage of screen real estate on a mobile device. Over time, as we explored the related work, we incorporated physical browsing research as a lens to look at our work, and thus led us to tailor our design to build upon these interaction approaches.

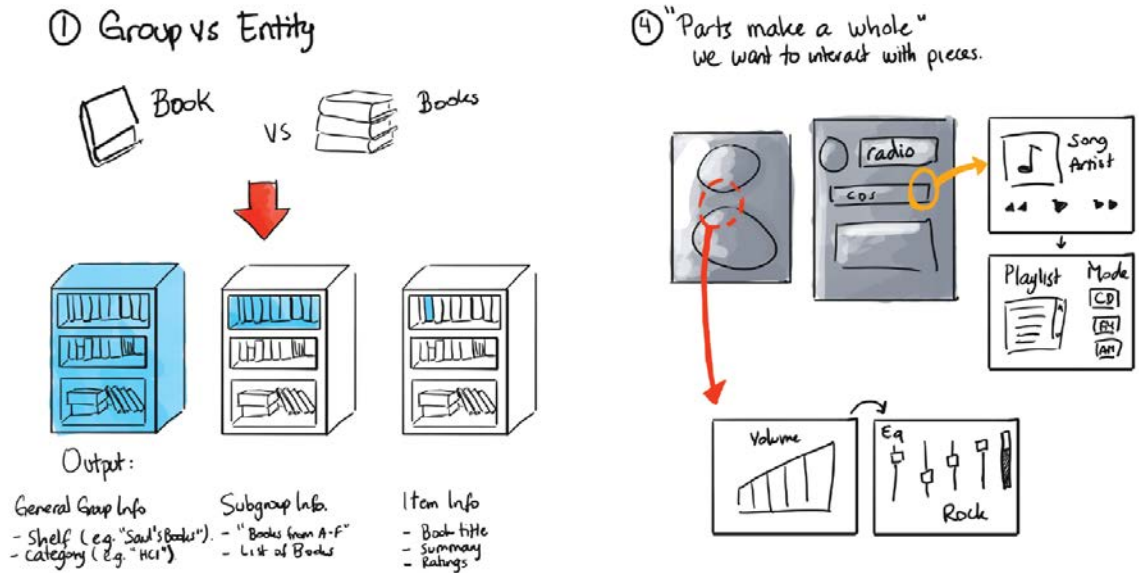


Figure A.3 Examples of early sketches in the design process.

A.5.3 First Ideation Stage

Our ideation process began with sketches that broadly explored how users may use a mobile device to present information from items in the physical world, this included controllable devices, but also explored other physical objects that could be digitally augmented such as books in a shelf. Some of these examples can be seen in Figure A.3. These sketches then became more focused and started exploring appliance interaction, as well as the change of the interface as a function of proximity (see Figure A.4). As the sketches evolved, some would turn into low fidelity prototypes such as storyboards and flipbooks. These low fidelity prototypes would also turn into medium fidelity prototypes as small, self-contained programs, which served to inform the implementation. This sketching process remained consistently throughout the design process as a thinking and communication tool.

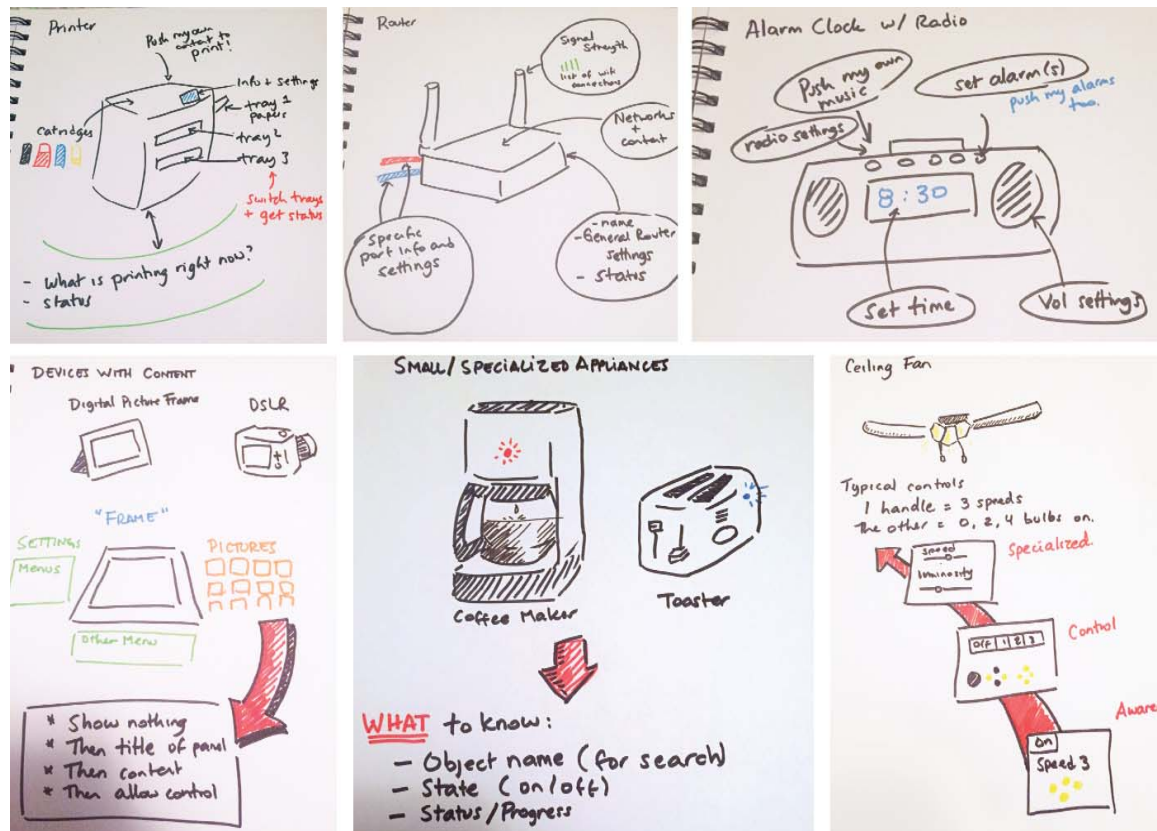


Figure A.4 Examples of sketches focusing on appliance interaction and proxemics.

A.4.5 Grouping and Classifying Items – First Framework

We grouped several sketches from the different stages and started to find certain themes that began to shape the framework. As shown in Figure A.5, the original framework had no separation between interaction and design constructs. It described an interaction flow and demonstrated how a set of interaction techniques fell into place within that flow. However, a lot of this initial structure is still

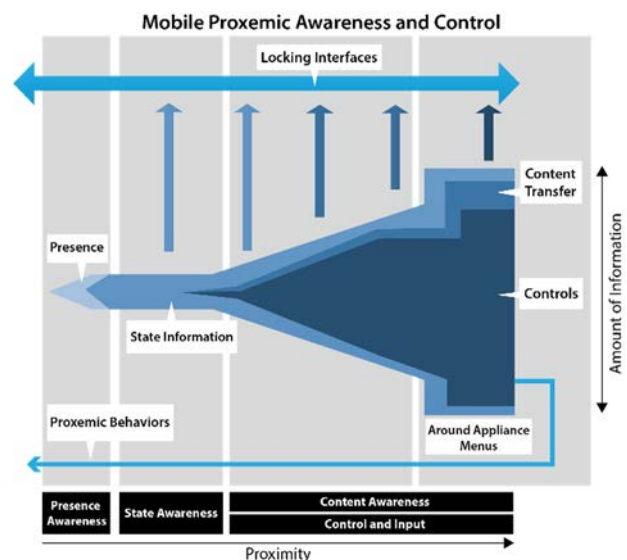


Figure A.5 First framework – mobile proxemic awareness and control.

present in our latest version, which illustrates presence, state and controls as gradual steps for revealing content about an appliance on a mobile device through proxemics. This distinction of constructs and design rules became more apparent as the process matured, and became especially clear with the creation of scenarios.

A.5.5 First Milestone

The first implementation focused largely on interaction techniques that fit into the interaction flow proposed. The lamp interface introduced the concepts of *gradual engagement* by showing increasing amounts of information and controls as a user approaches it; *locking* by preventing the information from disappearing when in close proximity; and *proxemic behaviors*, where the lamp would turn off as the user leaves the room as the result of an explicit action (tapping a checkbox on the interface to trigger the behaviors). The radio, on the other hand, showcased *around-appliance menus* as well as *information transfer*. Figure A.6 shows a picture of what the first milestone envisioned, controlling a desk lamp with a mobile device, where being in close proximity to the lamp reveals the most control possibilities.

As our first proof-of-concept implementation milestone was finished, we discovered that there were major issues with the real-world applicability, as well as aspects of proxemic interaction that needed to be reconsidered. The first item, which was caught during the early implementation process was that reorienting the mobile device caused the information and controls to disappear, often accidentally. This led us to the introduction of the locking mechanism. The next set of issues were discovered after our milestone had been reached after performing live demonstrations to the public. People would often move their device around the room and information would be suddenly revealed without them understanding why. This made us realize that discovery and selection of interactive appliances was a problem of its own and needed to be addressed, thus informing



Figure A.6 Visual of the first milestone.

our exploration and guiding our conceptual thinking. In a similar fashion, our third problem was the sudden transitioning without animations proved to be abrupt and confusing.

Another lesson gained from this implementation was that we could benefit from a better infrastructure to facilitate rapid

prototyping. This led to the development of the system architecture, discussed in Chapter 5. While the development did not lead to a toolkit, it facilitated our development of medium-fidelity prototypes, as eventually we were able to create individual appliance controls within a day. It also created a unified distributed model that could be understood by every client, which allowed us to use more than one computer and distribute the appliances throughout the room.

A.5.6 Expanding Theories and Informing the Framework

We decided to explore further theory that could help us target the new problems, which led to the presentation techniques portrayed in §3.2.3. Initially our theoretical explorations took us to information visualization to better understand presentation techniques for discovery and selection, and to augmented reality. The latter approach was later discarded. Furthermore, observing our prototype in action led us to reconsider what would be a feasible approach to proxemics to justify physically approaching an appliance. For instance, the first desk lamp example proved to be what we consider a deliberately simple example yet poor application of proxemics: a user can simply approach the lamp to turn it on or off. This case informed our design of future appliances (e.g. the new lamp being able to turn on or off at any distance), and eventually led to our more sophisticated

application of locking: the manual override including the proximity slider. Drawing from physical browsing research we were able to adapt our design to mimic already existing interactions.

As we incorporated new theories, our generic groupings would be further refined. The ideas were then regrouped by using personal constructs that defined a series of dimensions, which led to the appliance categorization in our framework described in §3.1. We then sketched different possibilities with the aim of conveying as many concepts as possible. In grounding our design, we can see that some of our implemented appliances are in fact based on the work presented in our physical browsing literature, such as the printer (Want et al., 1999) and the lamp (Seifried et al., 2009), which illustrates how we leveraged existing work to further identify our classification. At the same time, we turned back to the theory for guiding our designs, and attempted to explore how different interactions resulting from our hybridation could take place.

A.5.7 Augmented Reality Explorations

As this process was taking place, we also started experimenting with augmented reality prototypes. The expectation was that by having a visible camera image with controls overlaid on top, we would be able to better situate the context of actions. Our vision can be seen in Figure A.7 as a storyboard. We implemented three prototypes exploring target finding and evaluated these prototypes as part of a class project (CPSC 681 – Introduction to HCI Methodology). The end result were three different interfaces, two which made use of augmented reality (see Figure A.8). The goal was to find individual objects scattered around the room (as a form of discovery). The virtual targets could be associated to isolated physical objects, clusters of objects or physical locations. This study design was by no means a final study design, yet our observations led us to stop moving in this direction and return to non-augmented reality approaches. We can summarize our lessons as (1) reflections on the use of augmented reality and (2) the unpredictability of the overall system.

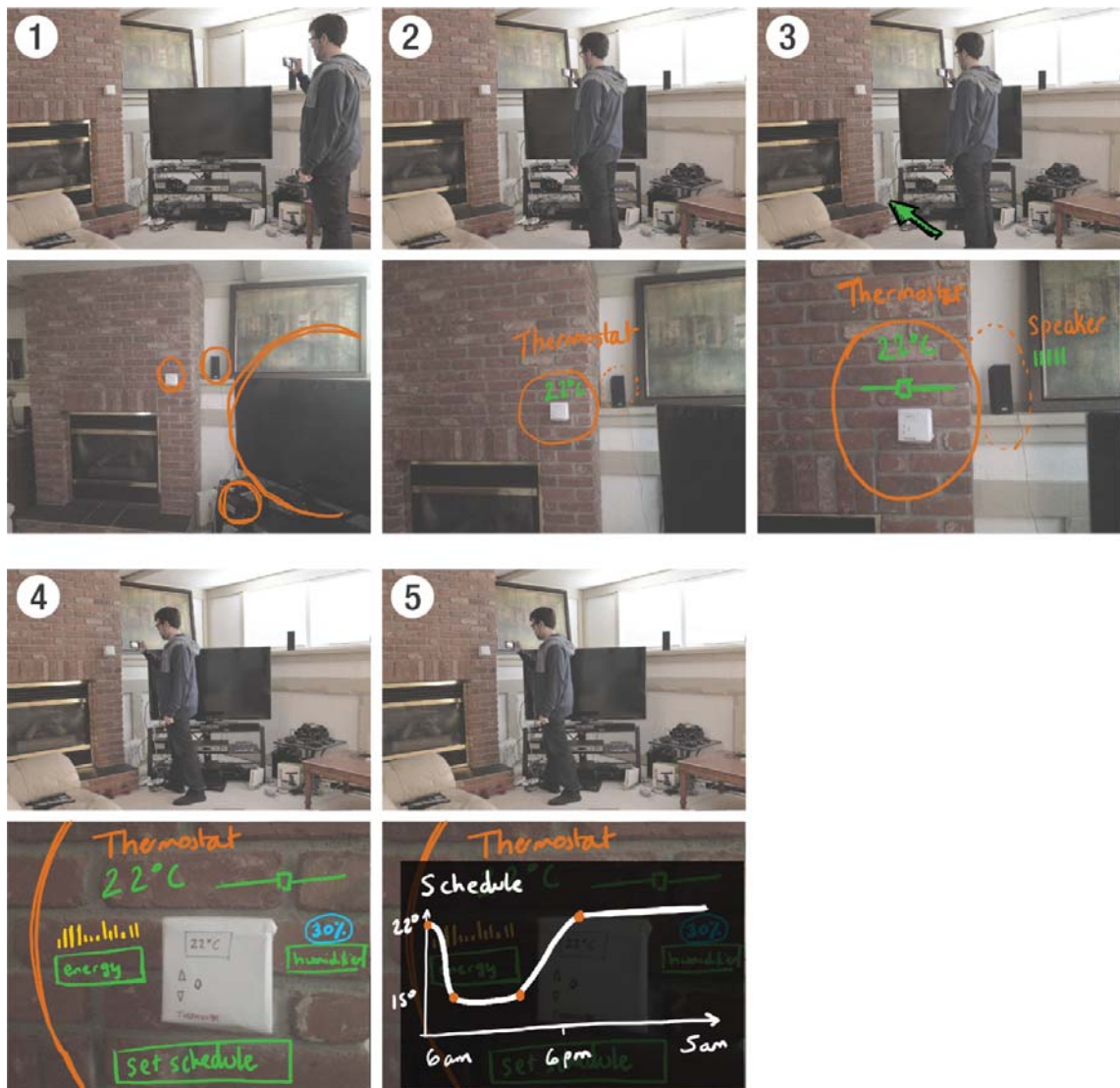


Figure A.7 Storyboard showcasing the interaction flow using augmented reality.

Reflections on the use of Augmented Reality. Applying augmented reality on a tablet for appliance interaction seemed very appealing at first. The idea that information could be overlaid on top of a physical object was intriguing, and we thought would make the interaction more situated, as opposed to having a separate interface from what is seen in the real world. However, one thing we found is that in our development of the augmented reality prototypes, viewing the camera image for extended periods of time would lead to dizziness. Augmented reality also proved to decrease the flexibility of the interfaces, as the camera image would always have to show the currently engaged object, which may be in the user's

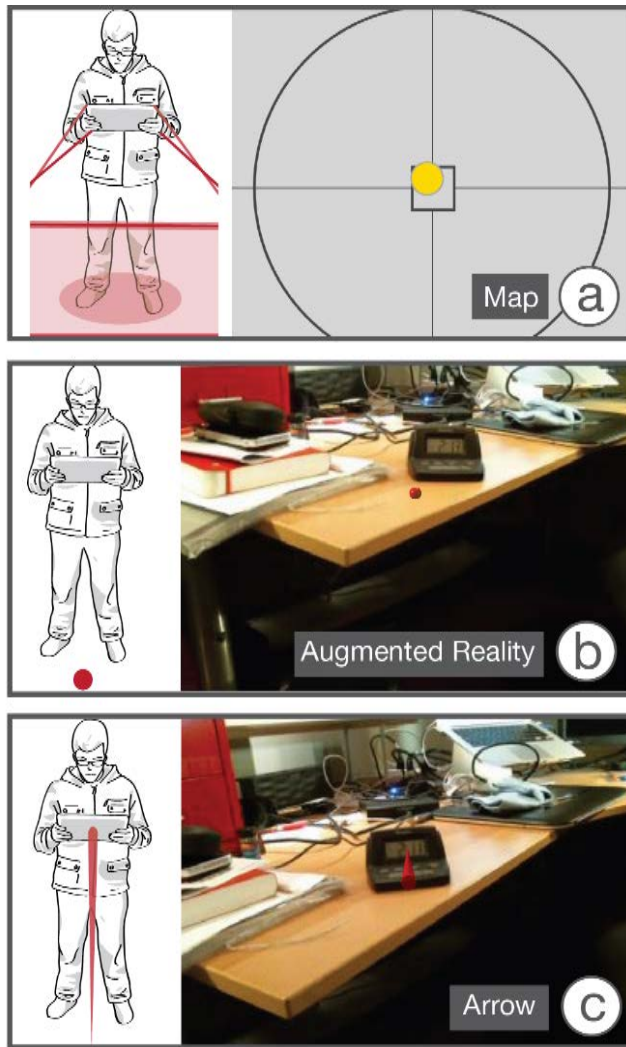


Figure A.8 Our implemented navigation techniques for target finding: (a) map, (b) augmented reality, and (c) arrow.

personal field of view, but not reflected in the tablet's camera image. This led us to believe that we could abstract the spatial interface view and show the engaged object on screen even if it does not exactly fall within the field of view of the tablet, thus allowing us to further expose the control interfaces. Having an abstracted view as opposed to a camera view like in augmented reality also meant that we could have more information and controls and prevent the visual clutter. Another interesting observation that we found during the pilots was that participants often fixated themselves on the tablet camera image and rarely looked at the physical world.

While further evaluation would be

required to test whether our current prototype allows for interaction without having users fixate solely on the tablet view, we found that in our personal experience the second milestone led us to look towards the appliances as a confirmation that the physical objects corresponded to the virtual objects and thus spend less time fixated on the camera. Furthermore, we believe that our use of animations in our next iteration facilitated the spatial navigation and reduced the need to constantly look at the screen.

Unpredictability of the overall system. We found that while the Vicon motion tracking system has up to millimeter accuracy, we were not able to predict when the tracking of objects would be lost. This loss of tracking could take place when a participant covers the tablet with their body and the cameras become unable to see it, or simply because of the noise caused by ambient light. When the tracking was lost, users would be required to leave the area and come back in, which interrupted the tasks and flow of the evaluation. There were also network inconsistencies in the building caused by interference on the wireless signal, which made the visual representations often lag, which is why we leaned towards internal sensing to create a more seamless interaction. While internal sensing along with animated transitions could mitigate some of the latency issues in the system, this did not solve potential issues caused by the loss of position data. We adjusted the cameras and created more stable markers, which reduced the instability but did not entirely solve it. For this reason, we believe that subjecting participants to a new study in the current conditions would be a highly frustrating experience, and the potential insight would quickly be overshadowed by tracking and network issues.

A.5.8 Towards the Final Implementation

With our new insight about augmented reality we decided to discard the approach and focus on spatial interfaces that are entirely digital. Some of the more mature sketches can be seen in Figure A.9, illustrating concepts such as grouping, proxies and overview. As these explorations were taking place, the framework was a lot more mature and established, and we had plenty of explorations that demonstrated a series of key concepts which we later turned into the different scenarios. At the same time, the architecture was also in place and allowed for more rapid implementations of controls for appliances. The code was flexible enough to enable us to do what we call software sketching: through small changes in the code we were able to explore different design variations and see them as a real-world application. Different components of these sketches can be seen in Chapters 4 and 5, such as the different kinds of overview as well as the variations in the different

Appendix – Methodology and Process

ways to position objects on screen. Thus we were able to quickly examine different solutions and see them work in an integrated manner with our prototype, leading to the system described in the previous sections. We have found a lot of benefit from the lessons learned throughout the process and can see how they came into play in the newest version of the system. The prototype is able to provide overview (discovery) of appliances in the room, enable for selecting an individual item, allow for viewing information about the appliance and enable control. The use of smooth animated as a function of distance and the internal sensing create a sense of smooth responsiveness (through feedforward and feedback) that allows users to

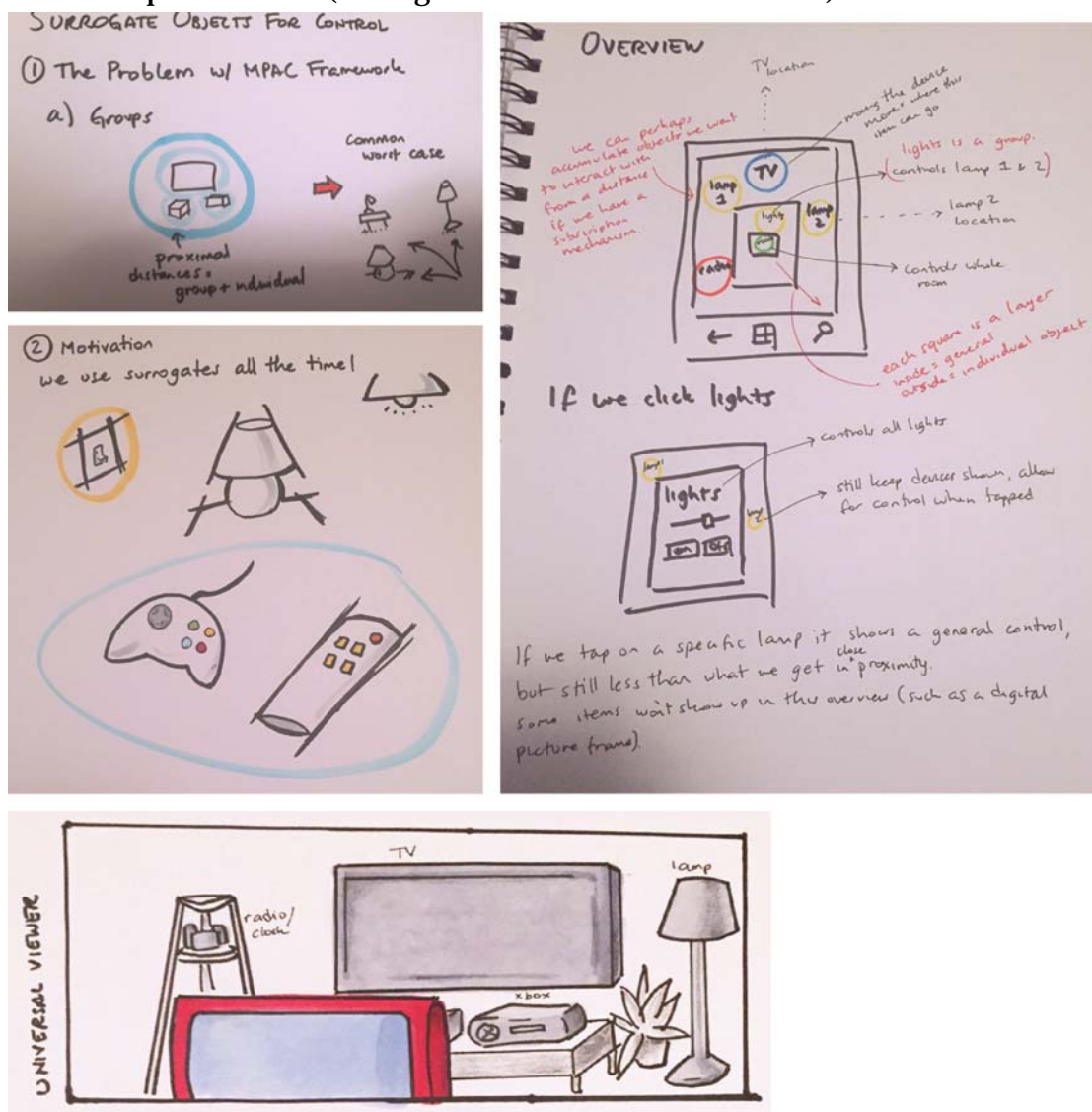


Figure A.9 Later sketches that informed the second milestone.

quickly understand the effect of the actions performed by moving in the space. At the same time, it is important to note that this is still a prototype – the tracking system is unstable and there is a significant loss of tracking that takes place, especially when approaching objects on the shelf, the network connection sometimes drops due to interference and there is still latency taking place when updating position information.

A.6 Discussion

As a result of the process and the methodology applied, we found that there were certain themes worth discussing. In this section, we describe our experiences with demonstrating our work to the public, the role of our personal experiences in the design process and our reflection on the effectiveness of the interweaving of different design approaches.

A.6.1 Live Demonstrations

We found that the sketches, together with small implementation examples were more than just a thinking tool. They also created interesting open-ended conversations during our live demonstrations. We found that sketches can be very visionary and thus leads people to be inventive and talk about relatable examples. For example, in one of the early conversations, one of the viewers portrayed his frustration when it came to losing manuals or product warranties, and discussed how they are items that are good to have, but not necessarily accessed often and thus may make sense to appear in close proximity to the appliance. When showing sketches together with the first milestone, people discussed the role of proxemics when it came to unreachable items, such as a ceiling fan. We also found that other non-HCI specialists made connections between our work and their domain of application, such as the value of integrating our technology to nursing homes and help the elderly find and interact with their appliances, as they often struggled to manually operate them.

Our first milestone was demonstrated at one of our laboratory demo days, which featured over 60 attendees. This allowed for quick focused discussions with industry partners, workers from the public sector and university staff. We found that unlike our concept design demonstrations, attendees would associate the implementation with a finished idea, and the feedback was centered specifically on the implementation and not the concepts being illustrated. However, we were still able to gain insight from these discussions, as we were able to find the fundamental issues with the interface discussed in the previous section. We found that in this context, real world applicability helps with refining the theory. This matches Gaines' (1991) progression in theory development – it allows us to understand what works and does not work which can inform future research through rules.

A.6.2 The Role of Personal Experience

Our personal experiences had a role in motivating some of our concepts. For example, one point of inspiration was a large printer located in the same floor as our laboratory, which requires people to leave the lab in order to pick up a printout. This printer has a small console on the side that was very difficult to navigate. It also has the capability of showing error messages for the failed printouts, but these are often difficult to read and do not suggest any potential actions that could solve the problem. For example, in some cases it would say: “tray empty”, and it meant that one particular tray out of four was out of paper. Yet, checking the main paper trays would show otherwise. The biggest issue was user's awareness of the printer issues took place once the printout was issued and the user had left the lab to pick up their printout. Seeing these issues inspired us to look into two concepts: first was the role of situated interactions, which were realizable through our use of proxemic interaction theory, and second was the spatial distribution of controls.

A.6.3 Interweaving of Design Approaches

The research showcased in this thesis demonstrates an instance of research through design in which the multiple approaches are combined. The

Appendix – Methodology and Process

methodological explorations of research through design are relatively new, and to the best of our knowledge there have not been many examples articulating the design process in the context of this kind of research. The work presented in this thesis is an example of an interweaving between the four different design approaches. We believe the combination of the approaches took place as a result of the types of activities performed: *design guidelines* formulated the knowledge, *concept design* triggered the ideation for the process; *generic design* structured our understanding of the space and *proof-of-concept design* showcased this understanding as a real-world application. This shows a constant exercise of analytic inquiry to produce and leverage knowledge while generating artifacts through application to ultimately inform knowledge, thus leading to a larger knowledge base (Owen, 1998).

It can be seen that by looking deeper into what we considered aspects that worked and did not work for each outcome in the different design approaches, we were able to further extend our understanding of the space. Our theory and sketches defined the framework, and the framework informed the design of the prototype. By observing the performance of the prototype in the real world, we were able to inform our framework and thus expand the theory for this particular context. This matches Gaines' (1991) progression in theory development – we can understand what works and does not work for the applications of proxemic interactions which can inform future research in this area and lead to well-accepted rules.

