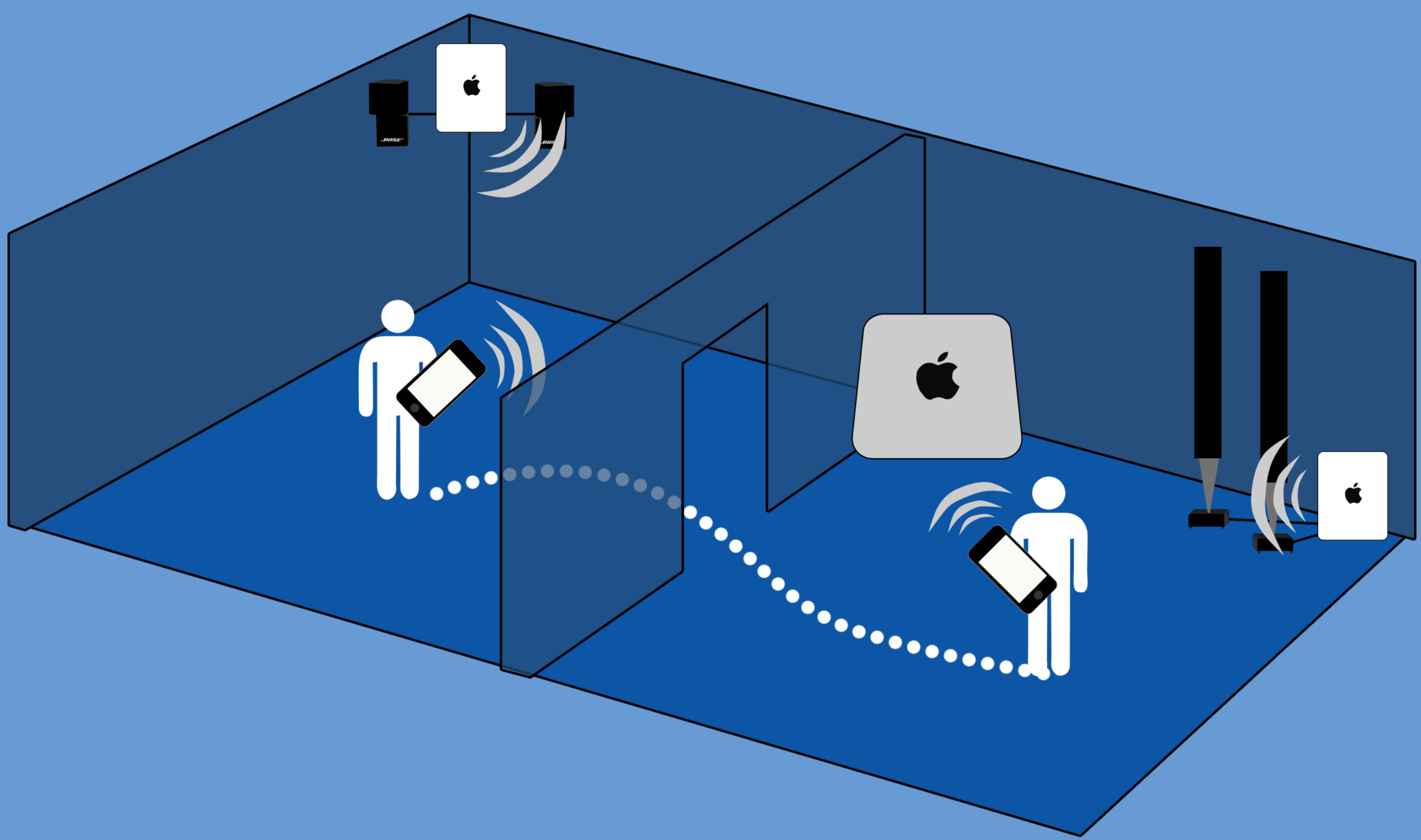
Software Engineering - Aalborg University 2012

AirPlayer

Proxemic Interaction in Multi-Room Music Systems



SW107F12
Master Thesis

Kasper Gadensgaard
Mathies Grøndahl Kristensen

Summary

The research goal for this master thesis was to explore how principles from proxemic interaction could assist users in the interaction with multi-room music systems. To answer this question, AirPlayer was developed. AirPlayer is a prototype multi-room music system where proxemic interaction principles have been incoorporated in the interaction design. This master thesis covers both the design and implementation of AirPlayer, which is based upon a technical framework developed during the the previous semester project. The reason for developing our own framework is that none of the commercial systems nor any of the available open source projects offered the technical features to include design based on proxemic dimensions in their multi-room music systems.

To evaluate the developed prototype, AirPlayer was tested for a three week testing period at two different households. The testing was conducted as a qualitative study in order to evaluate the usefulness of the implemented proxemic dimensions, Location and Movement, through an in-situ field study at the two households. Throughout the testing period, both families were interviewed while the system itself made use of extensive data logging. Combined, both the interviews and the log files formed the basis for an analysis regarding the utilization of proxemic interaction in a multi-room music system.

From the study it was found that the users appreciated the proxemic dimensions in the interaction with their multi-room music systems. This showed clear indications that the implemented proxemic dimensions assisted the users with the general tasks of managing their multi-room music systems. Furthermore, by analyzing the data from the log files, it was found that both households had a tendency to decrease the amount of direction interactions with the system while utilizing the Movement dimension within AirPlayer.

Aalborg University

Department of Computer Science

Title:
AirPlayer - Proxemic Interaction in Multi-Room Music Systems

Field:
Human-Computer Interaction

Master Thesis Period:
February 2nd 2012 - June 8th, 2012

Project Group: sw107f12
Kasper Gadensgaard
Mathies Grøndahl Kristensen

Supervisor:
Jesper Kjeldskov

Number of appendices: 3

Total number of pages: 82

Number of pages in report: 64

Number of reports printed: 4

Abstract:

The goal of this master thesis was to explore how proxemic interaction principles could assist users in the interaction with multi-room music systems. In reponse, AirPlayer was developed, which is a prototype multi-room music system that utilizes proxemic principles in the interaction design. This master thesis covers the design and implementation of AirPlayer, which is based upon a technical framework developed during the previous semester project. Furthermore, a qualitative study was performed to evaluate the usefulness of the implemented proxemic dimensions, Location and Movement, through an in-situ field study at selected households. During this testing period, interviews were conducted and system data was logged, which formed the basis of an analysis regarding the utilization of proxemic interaction in a multi-room music system. As a result, we found indications implying that the implemented proxemic dimensions assisted the users in general tasks of managing their multi-room music systems and revealed a tendency to a decrease of direct interactions with the systems.

PREFACE

This paper substantiates the result of Software Engineering group sw107f12's master thesis at the Department of Computer Science under the group of Information Systems, Aalborg University. The report is documentation for the master thesis, which proceeded in the period from 2nd of February 2012 until 8th of June 2012.

The master thesis was developed within the research area of Human-Computer Interaction and focusses on the topic of proxemic interaction in a ubiquitous music environment. The paper documents the development and field study of a multi-room music system which integrates the proxemic principles in the interaction design.

Unless otherwise noted, this report uses the following conventions:

- Cites and references to sources will be denoted by square brackets containing a number referring to an entry in the bibliography on page 65.
- When a person is mentioned as *he* in the report, it refers to *he/she*.

Throughout the report, the following typographical conventions will be used:

- References to classes, variables and functions in code listings are made in `monospace` font.
- Omitted unrelated code is shown as "..." in the code examples.
- Lines broken down in two are denoted by a ↵.
- Editorial notes in quoted citations from performed interviews are presented in square brackets.

The code examples in the report are not expected to compile out of context.

Appendices are located at the end of the report.

The master thesis was developed with help and guidance from our supervisor Jesper Kjeldskov. We would like to use this opportunity to thank him for his help and support. Also, a thanks to the test users who participated in the field-studies.

Aalborg, June 2012

Contents

Chapter 1

Introduction

With the rapid development within the field of ubiquitous music, a new culture for music consumption has arised. The continuous development of ubiquitous music devices has resulted in still smaller devices with higher capacity and a still richer feature set, and this development poses new challenges, especially within the field of Human-Computer Interaction (HCI). [17]

Recent technologies, in particular web- and cloud-based services, are providing users with global access to their music libraries. But, this development also opens for new possibilities for users to enjoy their music in a local ubiquitous music environment. For example, multi-room music systems combines global access to music with a network of locally connected ubiquitous music devices. With a multi-room music system, users are able to enjoy their music collection from any area of the domestic environment, that is, users can play different music in different rooms in the home, or play the same music synchronously in different rooms.

Another research area, which has recently received alot of attention within the field of HCI, is proxemic interaction. Proxemic interaction is a branch of context-awareness which accounts for the spatial relationships in the interaction between entities in an environment. The notion of proxemics was originally introduced by Edward Hall [13], treating physical space and distance between people in their interpersonal communication, however recent studies [12] [4] have tried to apply the same principles in the interaction between entities in a digital ecology.

In this master thesis, we seek to apply proxemic interaction principles in the interaction design of a multi-room music system, as defined by the following research question:

> *How can the principles of proxemic interaction assist users in the interaction with multi-room music systems?*

To explore the possibilities of proxemic interaction in a multi-room music system, the AirPlayer system has been developed. AirPlayer is a prototype multi-room music system, designed to support and include the proxemic principles in the interaction design. The work of this master thesis is based on the results obtained through the project of the previous semester as documented in the project report [10]. During the previous semester, an analysis was conducted to study several aspects regarding the area of multi-room music systems. From a technical study of existing systems and a range of user interviews, the requirements for AirPlayer were formed, and based on these, a technical foundation for AirPlayer was developed.

AirPlayer consists of two main components, i.e. a server application running on a PC and a client application running on a smartphone. The server application is the central component of the system, and implements the actual multi-room music system. It holds the users music collection and handles the individual zones in the system. The client application functions as a remote for the multi-room music system. From the client application, the users can control the music being played in each of the individual zones. The interaction through the client application is assisted by the proxemic interaction principles, and implements two of the five of the proxemic dimensions, being location and movement.

To get an understanding of how the principles of proxemic interaction effects the usage and interaction with a multi-room music system, AirPlayer was tested by a selected group of test participants. The tests were conducted as field studies, where the system was installed in the users natural home environment. After a short description of the test setup and an introduction to AirPlayer, the system was handed over to the test participants to use in their daily routine during the test period. Based on the results from these tests it was our goal to answer the overall research question.

1.1 Contribution

Through this project, the research question will be explored and answered. A prototype multi-room music system will be developed with support for the proxemic principles in the interaction design. The system will be deployed for testing in a real user environment by a selected group of test participants. While several previous studies have explored the possibilities for proxemic interaction in a ubiquitous computing environment, no research exists within the area of the music domain. Furthermore, only few contributions within the field have tested the interaction techniques in a real user environment.

The work of the master thesis is based on the results obtained through previous semesters project. During the last semester, a framework was developed to create the technical foundation of a multi-room music system with support for the proxemic interaction principles.

This report documents the work of the master thesis, namely the development of a functional prototype based on the technical framework, and the deployment the prototype in the home environment of the test participants to explore the possibilities with proxemic interaction in a multi-room music system. The test results were studied and analyzed in order to answer the research question of this master thesis.

CHAPTER 2

RELATED WORK

As stated, the focus of this master thesis was to explore the possibilities for applying the proxemic interaction principles to the interaction design of a multi-room music system. To get an insight and understanding of current research within the fields of ubiquitous music and the proxemic interaction principles, a selected range of contributions within these fields were studied.

Ubiquitous music is a relatively new area of research which has branched from the field of ubiquitous computing as a result of the technological advancement of digital music. As a new research area, ubiquitous music raises new challenges, particularly regarding the interaction design of ubiquitous music devices.

Proxemic interaction is the study of how spatial relationships between entities in a digital environment can be integrated in the interaction design. Research within the field of proxemic interaction seeks to apply an extended notion of proxemics to the interaction between people, digital and non-digital devices in a digital ecology.

Besides the scientific contributions, a range of multi-room music systems were studied to get an insight of technological possibilities within currently available systems. The study was not intended to cover the entire market of multi-room music systems, the goal was merely to gain a technological overview.

2.1 Ubiquitous Music

Ubiquitous music is a relatively new area of research within the field of HCI, combining ubiquitous computing and music-enabled devices in a networked digital ecology composition. Ubiquiotous music has originated from the well-established reasearch area of ubiquitous computing, as a reaction to the prevalence of technologies which have provided users with a global access to music through various sources, such as portable music devices and cloud based services [17] [20] [22]. AirPlayer applies this global access to music in a more local scope as a multi-room music system, capable of playing the users music collection from anywhere in the domestic environment. By utilizing a network of music devices, the user can access and play music from his music library from any room in his house, i.e. play the same music from the same source in different locations, or play different music from different sources in different locations in the domestic environment.

As a relatively new field of study, ubiquitous music still raises several unexplored research areas, especially within the field of HCI. As an example, Luciano Flores et al. [20] suggests, that ubiquitous music devices poses new challenges in the area of interaction design. The last decades of digital interaction design have been dominated by the classical WIMP design metaphor, which utilizes elements such as icons, windows, dialog boxes and scrollbars as tools for interaction, functioning as mediators between user actions and the objects being manipulated through the application. While the WIMP metaphor is widely available at a low cost, Flores claims that the metaphor presents several limitations for the interaction design, particularly in the context of ubiquitous computing and hence ubiquitous music, as the users are presented with a still increasing amount of information on still smaller devices. To account for this, Flores encourages not to be locked by traditional design metaphors, and instead base the application design on interaction rather than functionality. Flores et al. adopted this idea in the implementation of a series of sculptural sound installations called *Green Canopy*. Depending on the presence and spatial relationship between the sculpture and the surrounding people, the *Green Canopy* installations would provide the visitors with differentiated music experiences. In this way, people would interact with the sculptures merely by their presence and position in the environment instead of by direct input sequences. The interaction based design approach is also encouraged by Lars Erik Holmquist [14] who presents a design for a portable music device called *Sonic City*, where the interaction is based on the users environment. By using a range of sensors, *Sonic City* composes a playlist for the user. For example, a microphone takes in sounds from the environment, an accelerometer detects the users current speed, a metal detector registers changes in the environmental character. Based on input from these sensors, *Sonic City* composes a playlist suitable for the users current environment.

With AirPlayer we also seek to break with the traditional design metaphors in the interaction design of the application. Instead of depending on direct user input to control the music, we wish to base the interaction with the system on the spatial relationship between entities within the digital environment. This design metaphor was adopted through use of the proxemic principles to the define the interaction with AirPlayer.

2.2 Proxemic Interaction

The original conception of proxemics was introduced by Edward Twitchell Hall [13] as a field of research which *"identified culturally dependent ways in which people use interpersonal distance to understand and mediate their interactions with other people"* [12]. In relation to proxemic interaction, one of the more interesting parts of Halls contributions is his definition of the four proxemic zones, i.e. the intimate, personal, social and public zones. As the names imply, the shorter distance between the people is, the more intimacy and expectations of interpersonal engagement increases. According to Hall, people use the distance between themselves and their peers to match their social activities and also as a defensive mechanism to express if other people inappropriately intrude their more intimate zones.

While these proxemics is naturally applied in our everyday interpersonal interaction, it is not commonly expected in the interaction with digital devices in a ubiquitous computing environment. [4] Proxemic interaction in a digital environment has been the topic for several studies by Till Ballendat et al. [4], Nicolai Marquardt et al. [18], and Saul Greenberg et al. [12] among others. These studies are based on Greenbergs working thesis as stated below:

> *Just as people expect increasing engagement and intimacy as they approach others, so should they naturally expect increasing connectivity and interaction possibilities as they bring themselves and their devices in close proximity to each other and to other things in the ecology. [11]*

However, as Ballendat and Greenberg states, proximity in a ubiquitous computing environment differs from the original notion by Hall in that a mixture of both people, digital devices and non-digital devices are involved in the interaction with a given system, as illustrated in Figure 2.1.

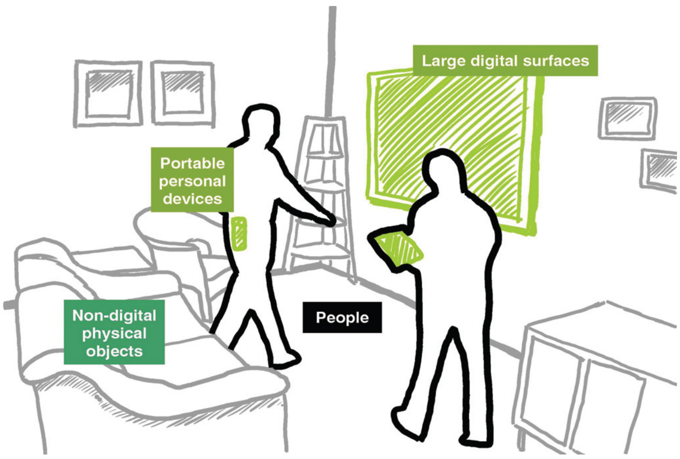


Figure 2.1: Proxemic Interaction in a Digital Ecology [11]

To accomodate for this they suggest to introduce an extended notion of the proxemic principles to monitor proximity in a ubiquitous computing environment, as defined below:

Distance decribes the distance between entities which is a fundamental property. Distance between entities can be described by a continuous measure in meters for example, or a discrete measure by introducing zones in defined intervals.

Orientation describes the relative orientation between entities. Orientation can be described by a continuous measures as the angle between one entity in relation to another or discrete, e.g. the entity is facing towards, somewhat towards or away from another entity.

Identity describes the identity of the entity itself. This information can consist of the type of the entity and its attributes to uniquely identify the entity, but also be limited to only containing a way to distinguish one entity from another.

Location describes the physical location of an entity within the environment. The location is not only dependent on the position of an entity, but also the contextual properties of the environment.

Movement describes a change in the distance, orientation and location of an entity over time. The movement can be described by a continuous measure such as a vector representation of orientation and speed, or by a discrete measure, e.g. the entity is turning towards another entity, or an entity is moving away from another entity.

Individually, each of these dimensions describe a proxemic property of a digital environment, and together the dimensions can be combined to enable a multitude of both implicit and explicit interaction methods in a digital ecology. Greenberg points out that one or several of these dimension have previously been implemented in a ubiquitous computing environment, however only a very few have accounted for all of them.

For AirPlayer we wished to examine the possibilities with each dimension of the extended notion of the proximic interaction principles. We sought to explore the potential of the individual dimensions and the goal was to develop a set of design ideas for each dimension to study the possibilities with the interaction in its pure form.

Based on the extended notion of proximity, Nicolai Marquardt et al. [19] developed The Proximity Toolkit. When capturing proximity data from an environment, a wide range of equipment is available, including a variaty sensors, vision analysis tools, motion capturing tools and so on. According to Marquardt the results acquired from reading from these sensors is always a trade-off between between factors such as data accuracy, equipment cost and development time to process the gathered data. The Proxemity Toolkit is based on the Vicon Motion Capture system which is capable of detecting precise motion of marker tags within an environment. From the received results from the sensor, the toolkit provides an event-based approach and allows developers to disregard some of the technical aspects of interpretting sensor readings from the environment. In this way, developers can focus on designing the proxemic interaction for their application while technical aspects regarding reading and processing sensor input can be omitted. The toolkit provides fine-grained proxemic information about the environment, i.e. people, digital and non-digital devices and their spatial relationships by the use of the five proxemic dimensions defined in the extended notion, which are distance, orientation, identity, movement and location. By use of The Proxemity Toolkit, developers can rapidly create prototype applications to explore the possibilities for proxemic interaction while disregarding low-level programming details. Figure 2.2 illustrates a developer working with The Proximity Toolkit. The toolkit provides an abstract representation of the environment and its entities.

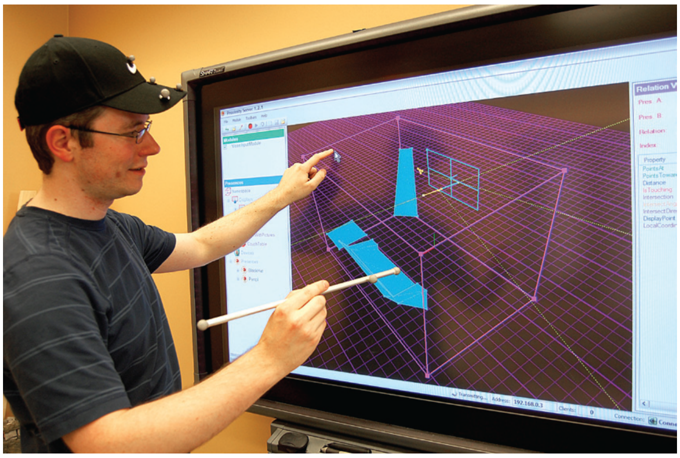


Figure 2.2: A developer experimenting with The Proximity Toolkit [19]

While The Proximity Toolkit provides all the required functionality for experimenting with the proxemic interaction principles in AirPlayer, it is primarily targetted interaction with large digital display surfaces. Furthermore, it depends on the Vicon Motion Capture system, which is a relatively expensive solution, and with a limited budget, a cheaper solution is required for AirPlayer. For this purpose, a simple implementation of WiFi triangulation will be used. By using signal strength results from the users smartphone, the system can approximate the users position within the environment. While this implementation poses some limitations to the proxemic dimensions, i.e. Wifi triangulation provides no information regarding the orientation of entities within the environment, it is sufficient to create a functional prototype to explore the possibilities with proxemic interaction.

A later group of studies experimented with the utilization of proxemic interaction by implementing a series of prototype applications based on The Proximity Toolkit. *ViconFace* is such an application developed by Rob Diaz-Marino et al. [9]. The application features an animated cartoon face on a large display which tracks people in the environment. By using the proxemic dimensions, the cartoon face responds to the proximity of people in the environment, both visually and verbally. For example, the eyes of the face follows people as they move around the room. If people face away from the face it simulates sadness and if people are facing the face it simulates happiness. Similarly based on The Proximity Toolkit, Greenberg et al. [12] developed a series of applications including *Proxemic Media Player*, *The Proxemic Presenter* and *Proxemic Pong*. The above applications have in common that they have been primarily developed with the purpose of demonstrating capabilities and utilizations of The Proximity Toolkit, however later research by Adrian Clark et al. [7] based on Greenbergs results, have studied the utilization of proxemic interaction in a real user environment. Clark developed a media player application which combined the proxemic interaction principles with direct input, voice and gesture based interaction with a large display. From the distance the users could interact with the display by using voice or gesture input on a secondary display, and within an armslength the users could interact directly with the display by using touch input. The application uses the proximity of people in the environment to vary the level of detail of the information displayed on the screen. At close range, the screen would show a large amount of information, and the user could use the touch interface of the display to interact with the application. As the user moved away from the display, the amount of information displayed on the screen was reduced to compensate for the reduced accuracy of the user's pointing input. This allowed the user to indirectly interact with the display from the mid-range distance. If the user would move still further away the amount of information on the main display would be further reduced, leaving the pointing input interaction less usefull. Instead, the user could interact with a secondary display using gestures or voice input. The application was implemented as a prototype and tested by a selection of test users. Based on the user reviews, Clark found, that the users generally appreciated the proxemic awareness of the application, and the proximity based pointing input performed well for general tasks at mid-range distance, however when controlling elements which required higher level precision, such as adjusting the volume or performing seek operations, the users preferred a direct input interaction, such as touch or gesture based interaction.

Based on Clarks research, we wished to combine the proxemic dimensions with direct input in the interaction design for AirPlayer. With a direct input interface, users gain a higher level of control over the playback of music. By using the touch interface on a smartphone, users can browse their music collection, create playlists, adjust volume and so on. Concurrently we sought to implement support for proxemic interaction within the application. The implementation of the proxemic interaction principles were to support the users in general tasks regarding control of a multi-room music system, such as selecting which music should be played where, or have the music move from one zone to another as the users moves to a different location.

2.3 Existing Systems

A multi-room music system is a system which is able to play music from the same source in multiple rooms in a home environment. In the study of existing systems, a collection of four different multi-room music systems from different vendors were examined, comprising *Bose Link* [6], *BeoLink* [5], *Sonos* [21] and *AirPlay* [2].

Bose Link is a multi-room music system integrated in Boses Lifestyle home entertainment system series. The system can stream local media from the Lifestyle system or stream media from external sources connected to the system. Bose Link does not support zoning in the general sense, but has support for two simultaneous audio streams that can be streamed independently from any of the connected sources to compatible receivers. Bose Link requires a dedicated remote control in each of the connected rooms to control the audio stream. An example Bose Link setup is illustrated in Figure 2.3. A LifeStyle media center is connected to a Bose Roommate and a SA-3 Amplifier as well as a Bose Link AL8 wireless audio link which streams wirelessly to a remote HomeWide system.

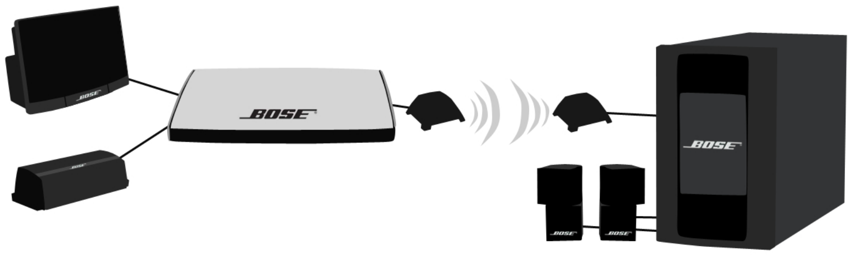


Figure 2.3: An example Bose Link setup.

BeoLink is a multi-room music system developed by Bang & Olufsen (B&O) which allows for streaming of media between B&O devices around the house. The system streams media from sources attached to a central entertainment system, or from sources available through the local network. In BeoLink, each room linked to the central entertainment system represents a zone. Each zone can play different media from different sources, or play the same media synchronously. Media playback in each zone can be controlled by direct input through a locally connected BeoLink eye or through a dedicated remote control. Also, B&O have implemented an app for both iPhone and Android which allows the system to be controlled from the users smartphone. An example BeoLink setup is illustrated in Figure 2.4. A BeoVision 10 and a BeoSound 5 located in the main room is connected to a BeoLink Wireless which streams the media wirelessly to linked rooms. An additional Wireless with an attached BeoLink eye for local control is connected to a set of BeoLab 8002 speakers.

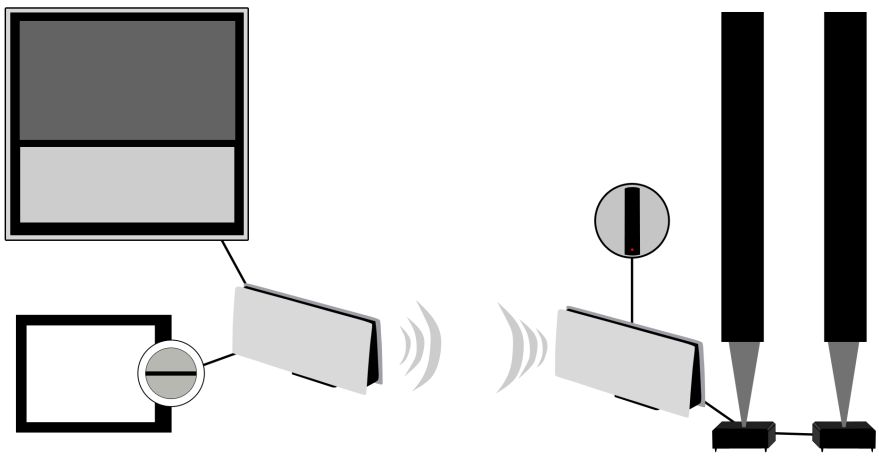


Figure 2.4: An example BeoLink setup.

Sonos is one of the leading products on the market of multi-room music systems which allows for streaming of audio between compatible devices around the house. The system streams media from locally attached sources, or from sources connected through the local network. Each player in a Sonos system represents a zone, and each zone can stream different media from different sources, or zones can be grouped to stream the same media synchronously. Each zone is controlled remotely from a Mac, PC, tablet, smartphone or the dedicated Sonos Control. An example Sonos setup is illustrated in Figure 2.5. The setup features a Play:5, Play:3, Zoneplayer 120 and Zoneplayer 90, all connected wirelessly to each other sharing the same music sources.

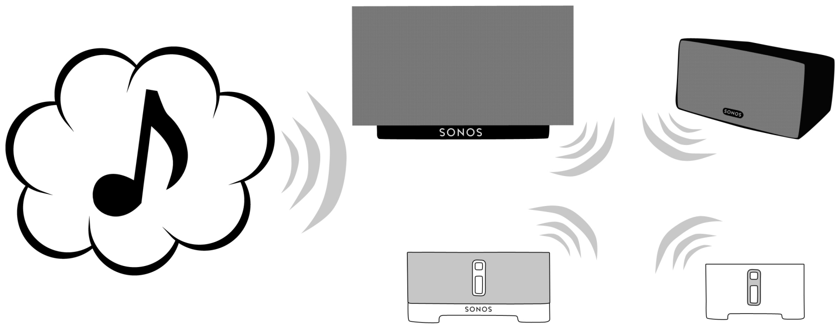


Figure 2.5: An example Sonos setup.

AirPlay is a wireless streaming protocol developed by Apple, and allows streaming of media between compatible devices around the house. The system steams media from iTunes running on a local computer, or from an iPod, iPhone or iPad connected to the local network. AirPlay does not support zoning directly, but is centered around a number of independent players, i.e. from a compatible player, users can stream media to one or more compatible receivers. In this way, each player with connected receivers practically represents a zone in the system. Each zone is controlled locally from the direct input interface of the player. If music is streamed from iTunes running on a local computer, the playback can be remote controlled from an iPod, iPhone or iPad. An example AirPlay setup is illustrated in Figure 2.6. Audio is streamed wirelessly from compatible sources, i.e. a Mac computer running iTunes, an iPad and an iPhone to compatible receivers, i.e. an Apple TV and an AirPort Express.

While the study of existing systems did not cover the entire market of multi-room music systems, the study of the systems provided an insight in the available technologies on the market. All of the studied systems are classified as multi-room music systems and supports playback of music in multiple locations in a domestic environment, however none of the studied systems had native support for proxemic interaction and none of the systems provides the possibility to implement alternative interaction methods. As a response, a technical platform for AirPlayer was developed from scratch with support for proxemic interaction. The platform is based on the AirPlay protocol which allows streaming of music via Wifi to compatible receivers. AirPlay features seamless wireless installation and integration with the users existing music systems. This property eases the deployment of the system in the homes of the test participants. Furthermore, AirPlay is an inexpensive solution compared to competing products, and the system can be extended with an additional zone by just adding an AirPort Express,

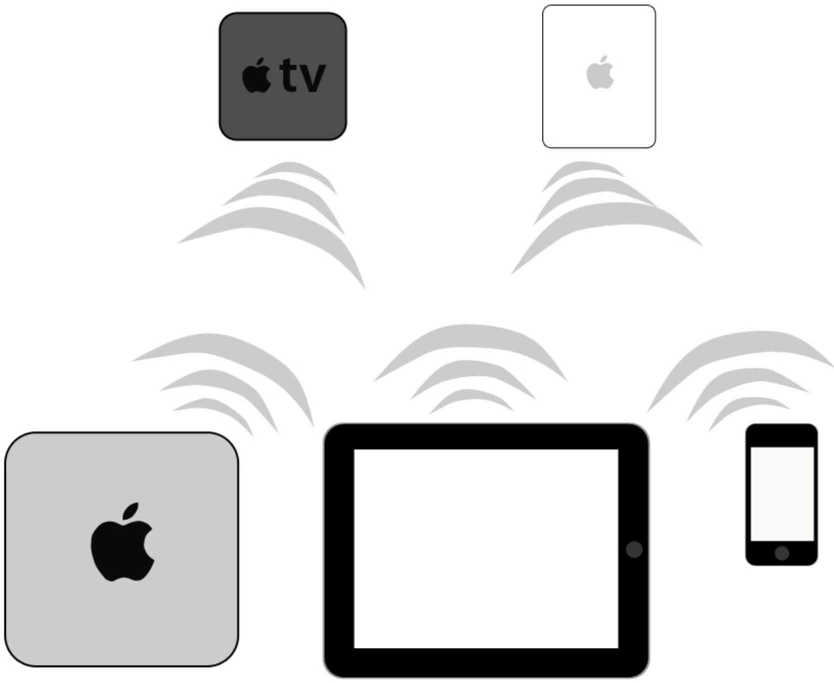


Figure 2.6: An example AirPlay setup.

resulting in an effective cost less than $100 per zone (not including the users existing music setup) [1]. The AirPlay protocol is proprietary, and only licensed applications are allowed to stream to compatible devices, however, the protocol have been reverse engineered by Jon Lech Johansen, and the private RSA key was released in an open source application called JustePort [15], effectively allowing streaming from third party applications. Based on Johansens release, the technical platform for AirPlayer was built on top of the AirPlay protocol. Details regarding the implementation of the technical platform was covered in previous semesters project report [10].

2.4 Summary

With the development within the field of ubiquitous computing, in particular ubiquitous music, a new culture of music consumption has arisen. Users are presented with still more information on still more feature rich devices which still decrease in size. This poses new challenges, particularly for interaction design within the field of HCI. Where traditional design metaphors such as the WIMP metaphor poses several limitations for interaction design, recent studies suggest application design based on interaction over functionality.

Proximity is the interpretation of spatial relationship in the interpersonal interaction between people and was originally defined by Edward Hall. Recent studies have extended Hall's to the interaction in a digital ecology, where distance, orientation, identity, location and movement define the spatial relationship between digital and non-digital devices and people. The extended notion of proximity was implemented in a toolkit called The Proximity Toolkit, which provided an event-based abstraction over low-level programming details, allowing developers to explore the possibilties with proxemic interaction design. The Proximity Toolkit was the base for several studies, studying the possibilities of the proxemic dimensions and the usability of different proximity based interaction designs in practice.

Besides the scientific contributions, a range of existing multi-room music systems were studied, comprising Bose Link, BeoLink, Sonos and AirPlay. The study of existing systems provided an insight in the possibilities with existing technologies on the market. None of the studied systems had support for proxemic interaction, and none of the systems allowed for third party remote applications to interact with the systems. As a response, a technical framework was developed from scratch, as described in the report of the previous semesters project [10].

CHAPTER 3

DESIGNING PROXEMIC INTERACTION

The study of related contributions within the field of proxemic interaction formed the basis for generating ideas for the proxemic dimensions in the interaction design of AirPlayer. Ideas were generated for each of the five proxemic dimensions, i.e. distance, orientation, identity, movement and location. The interaction design ideas will focus on each isolated dimension in its pure form.

This chapter contains the concept ideas for each of the five proxemic interaction dimensions in their pure form, that is, propositions for interactions which are solely based upon the individual dimension without being dependent on any of the remaining dimensions. Even though it is possible to combine the dimensions in order to achieve the desired way of interaction within a given setting, studying the dimensions in their pure form enables a different way of thinking and perceiving the possible interactions.

The purpose of this chapter was to discover new ways of interacting with multi-room music systems based on the proxemic dimensions. After the five dimensions have been described and ideas have been derived, a discussion regarding the potential of the ideas is presented. This discussion was used to decide which of the proposed ideas was to be integrated within the interaction design of AirPlayer.

3.1 Distance

Distance describes the distance between entities and can be described by a continuous measure in meters for example, or a discrete measure by introducing zones in defined intervals.

In relation to multi-room music systems, we suggest that using the Distance dimension the remote control of the multi-room music system could provide users with a differentiated UI. That is, the users should be presented with a user interface depending on the given distance to a given zone. For example, if a user is in close proximity to the center of the zone, i.e. the Airport Express device, an advanced controls user interface, used for configuation purposes, could be presented. In contrary, if the user is merely present in the zone, or adminstrating the zone while being inside a different zone, only the basic controls for playing, pausing and so on are presented to the user. This interaction is illustrated on Figure 3.1.

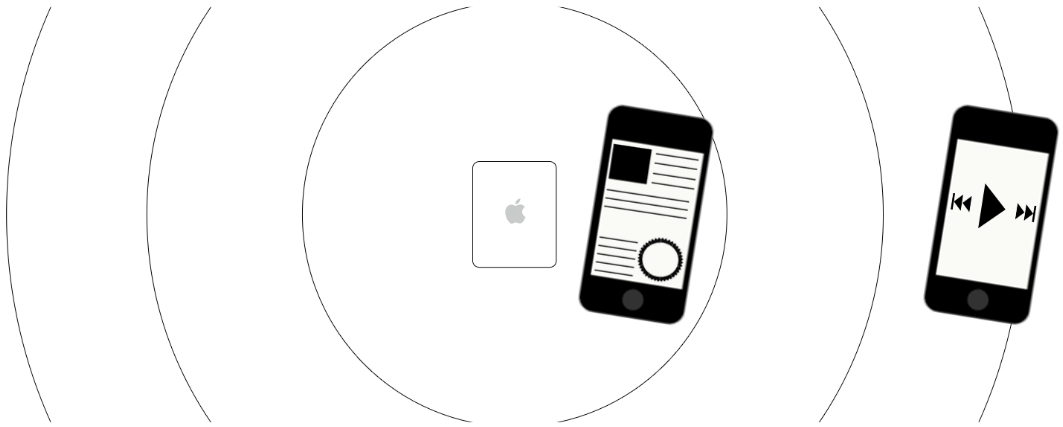


Figure 3.1: Differentiated interface based on distance.

In the illustrated example, the remote closest to the center of the zone, i.e. closest to the AirPort Express device, presents advanced control of the system while the remote furthest away presents only simple control of music playback. This effectively limits the need of having multiple devices known from commercial systems to manage different roles within the system by merging two functionalities into one application.

3.2 Orientation

Orientation describes the relative orientation between entities. Orientation can be described by a continuous measures as the angle between one entity in relation to another or discrete, e.g. the entity is facing towards, somewhat towards or away from another entity.

This property could be used in the context of a multi-room music system to introduce a higher level of control when using the remote control. That is, by introducing a set of rules based upon the orientation between the remote controlling device and the various installations throughout the households, it would be possible to utilize the Orientation dimension to decide which room is currently being controlled.

Another application would be that when pointing a remote control towards the stereo in a given zone, zone specific controls, such as volume, play and pause, would be presented. When pointing the remote away from the stereo, system wide controls would be presented, such as administration of playlists and songs in the music library as well as zone administration, as illustrated in Figure 3.2.

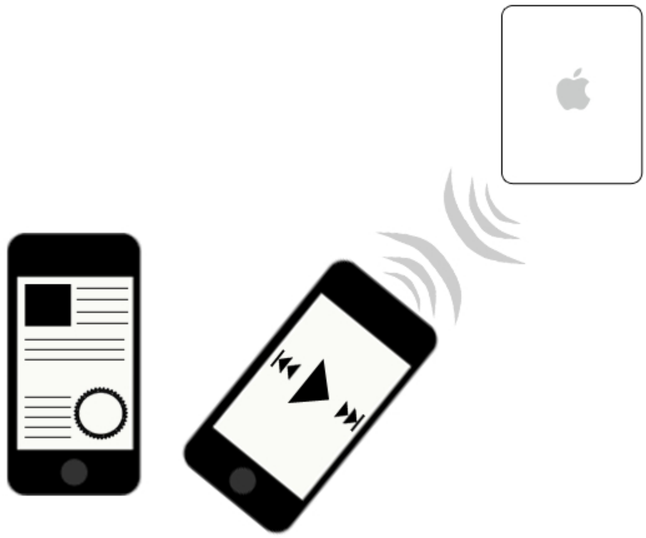


Figure 3.2: Differentiated interface based on orientation.

In the illustrated example, the remote pointing towards the stereo, i.e. towards the AirPort Express device representing the zone, presents controls for music playback for that particular zone, while the remote pointing away presents advanced system controls. Using either of these two proposed suggestions, would also limit the need of having multiple remote controls, as every remote would be able to control each zone, but also provides a differentiated display used to control both zone- and system wide settings.

3.3 Identity

Identity describes the identity of the entity itself. This information can consist of the type of the entity and its attributes to uniquely identify the entity, but also be limited to only containing a way to distinguish one entity from another.

By introducing Identity within a multi-room music system, it would be possible to distinguish users of the system from one another. This means that each user would be treated as unique entities within the system, each with their own music collections, playlists and personal settings. This would also enable users to be associated with individual user interfaces, effectively creating a personal profile for the entire multi-room music system.

Furthermore, each zone could also be described as unique entities. This means that each zone could act differently based upon factors such as the type of stereo, the size of the room and the room decorations. For example, when the clock is 8.00 PM in a household, the multi-room music system could disable the zones located within the children's rooms in a household. Furthermore, it could also adjust the maximum volume in the zones in close proximity of these rooms as the children are not to be awoken by loud music.

3.4 Movement

Movement describes a change in the distance, orientation and location of an entity over time. The movement can be described by a continuous measure such as a vector representation of orientation and speed, or by a discrete measure, e.g. the entity is turning towards another entity, or an entity is moving away from another entity.

By using the Movement dimension in the interaction design, it would be possible to make the music follow the user as he moves in between the zones in the househould. Effectively this means that the system is able to anticipate where the user is going, thereby preparing the music in the zone(s) ahead. Similarly, the system would also be aware of which zones the user is moving away from, thereby shutting down zones which are already playing as the user leaves the zones, as illustrated in Figure 3.3.

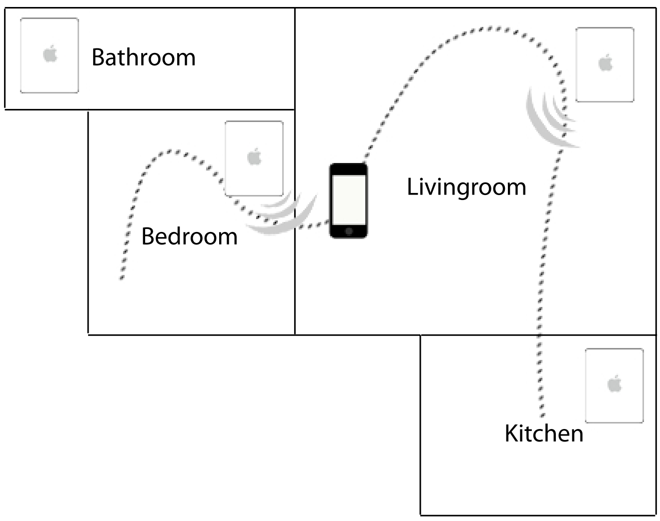


Figure 3.3: The music follows the user around the house based on the users movement.

In the illustrated example, the user starts the music in the bedroom of a household. The user then walks towards the living room, where the system starts the music just before the user enters the room, creating a fluent transition to the next room. The living room will now be playing the same music, at the same volume at the exact some position as the bedroom. Furthermore, as the user progresses further into the livingroom, the system will halt the music in the kitchen, which leaves the living room as the only zone currently playing music. The user will experience that whereever he goes in the household, the closest zone in proxemity will play the specific music and playlist that he started out with.

3.5 Location

Location describes the physical location of an entity within the environment. The location is not only dependent on the position of an entity, but also the contextual properties of the environment.

In a multi-room music system, the location dimension could be used to introduce location awareness in the system. Specifically, it would open the possibility of having remote controls which are aware of their positions, i.e. the zones, in which they are located, thereby automatically selecting and controlling the closest zone based upon the location of the remote controlling unit, as illustrated in Figure 3.4.

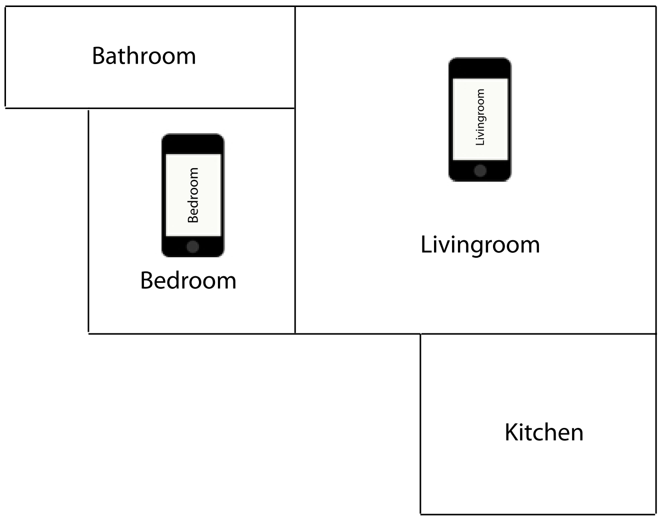


Figure 3.4: Automatic selection of closest zone based on location.

In the illustrated example, the remote applications have automatically selected the closest zone to be controlled, i.e. the beedroom and livingroom zones respectively. This type of interaction based on the Location proxemic dimension would aid users in selecting the closest zone, thereby potentially reducing both time and confusion in relation to selecting the zone in which the user is currently present.

3.6 Choosing Proxemic Dimensions for AirPlayer

The two proxemic dimensions, Location and Movement, were selected for the AirPlayer system to inspire new ways of interaction in terms of multi-room music systems. Below are the ideas and reasoning for choosing these proxemic dimensions.

Location

Using the Location dimension when designing new ways of interacting with a music system would introduce the possibility of selecting the closest zone in the AirPlayer system. The analysis from the previous semester showed that commercial systems either use multiple remote controls distributed throughout the zones of the household or lacked the feature of automatically selecting the closest zone within the system. This feature was therefore also selected as a proxemic dimension which were to be implemented in the AirPlayer system.

Movement

By designing and implementing interaction forms based upon the Movement dimension, it would make it possible for the music to follow a given user around in the household. This means that by implementing the Movement dimension in the AirPlayer system, the system should be able to anticipate the movement of a particular user, thereby creating a different way of interacting with a multi-room music system.

These two dimensions were implemented in the interaction design of AirPlayer. While the proposed design ideas for all the dimensions would be suitable for the study of proxemic interaction in a multi-room music system, only these two were selected, primarily due to time constraints of the master thesis. The argumentation for possible inclusion of the last proxemic dimensions, i.e. distance, orientation and identity are elaborated below.

Distance

By introducing the Distance dimension in the AirPlayer application, it would be possible to provide the users with a differentiated user interface, allowing them to change between basic and advanced controls based on distance to the center of the given zone. This could potentially be a useful feature which would limit the amount of remote controlling devices and/or provide the users with additional flexibility, however administrative features already exist in commercial systems.

Orientation

Orientation would introduce a higher level of control by using the orientation of the remote control in relation to the set of zones within the system to control the zone which is being pointed towards. Due to shortcomings in the selected platform however, it is not possible to derive the orientation of the remote controls in AirPlayer.

Identity

The proxemic dimension Identity would provide the possibility to uniquely distinguish both remote controls and zones within AirPlayer. AirPlayer already contains zones which can be uniquely identified using their zone properties, each containing a zone name, the volume level, a music queue etc. By introducing individual user identities, it would be possible for each user to have access to a specific subset of the entire music library as well as have their own favorite playlists associated with them. These features are not new in the context of neither single setup music systems nor multi-room music systems, and were therefore not incoorporated into AirPlayer.

3.7 Summary

Based on the study of related contributions, ideas were generated for implementing the proxemic dimensions in the interaction design for AirPlayer.

Ideas were generated for each dimension in their pure forms, i.e. Distance, Orientation, Identity, Movement and Location. Each of the generated presented new forms of interaction with a multi-room music system based on the proxemic dimensions.

From the generated ideas, two ideas were selected for implementation in AirPlayer, i.e. Location and Movement. While all ideas proved to be suitable canditates for exploring the possibilities with proxemic interaction, only these two were selected, partially due to timing constraints of the master thesis and partially because of shortcomings in the selected platform.

Chapter 4

AirPlayer

Through this master thesis, AirPlayer have been developed to explore the possibilities with proxemic interaction in a multi-room music system.

In this chapter, an overview of the system is presented, which outlines the functionalities of AirPlayer and describes the individual components of the AirPlayer system

Then a description of the interface design for the direct input interface is presented, together with a presentation of screenshots of the implemented design.

Next, a selection of implementational details are elaborated, covering central aspects of the implementation of AirPlayer.

Lastly, a set of enforced limitations within the system are presented as a reaction to the selected platform for AirPlayer. These limitations are introduced in the system to account for shortcomings in the selected protocol, in order to ensure a continuous and steady experience when using the system.

4.1 Overview

AirPlayer is a prototype multi-room music system with integration of the proxemic interaction principles in the interaction design.

As a multi-room music system, AirPlayer can play music in multiple rooms in a domestic environment. Each room connected to the system is represented through the abstraction of zones, and AirPlayer can play music in any number of zones simultaneously. Each zone can be controlled individually, idependently from any other one in the system, or zones can be grouped to allow synchronous control of multiple zones as well as synchronous music playback.

AirPlayer consists of two main components, i.e. a server application and a mobile client application. The server application is the core of the system and runs on a centralized server. The server contains the users' music collection and provides functionalities for the user to manage the collection, e.g. reorganize the music or create playlists. The management of the users' music is handled through iTunes [3]. This provides a familiar interface for the user to manage his music collection, and results in a simpler implementation on the server side of the application. The client application runs on the users smartphone and functions as a remote control for the system, which provides an interface for the user to interact with the system. Through the direct touch input interface of the smartphone, the user can control the music playback in each of the available zones, e.g. queue music on a zone, play and pause the music, adjust the volume and so on.

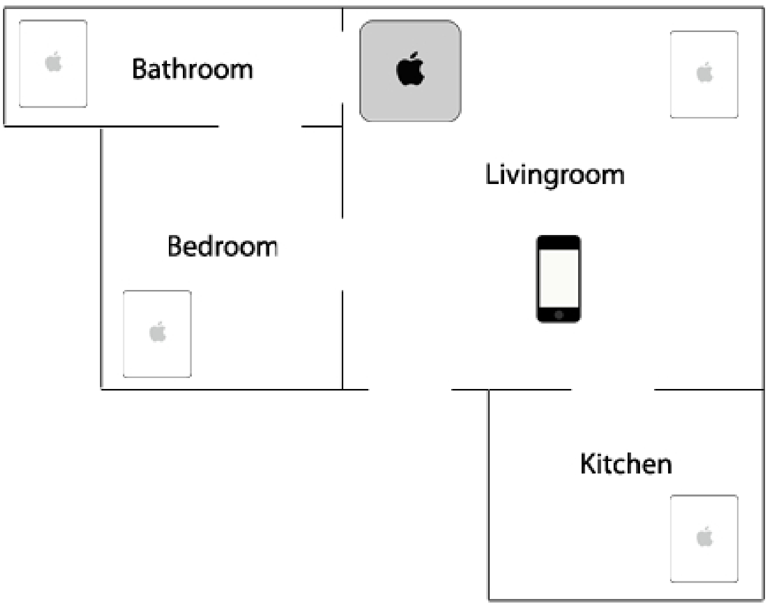


Figure 4.1: Illustration of the AirPlayer multi-room music system.

Besides the direct input interface, AirPlayer has support for proxemic interaction. Based on the spatial relationship relatively to the user, the user can interact with the system indirectly by proximity. The prototype of AirPlayer has support for two of the five proxemic dimensions, i.e. Location and Movement. Based on the user's current location within the environment, AirPlayer can present a remote interface for the current zone on the user's smartphones, providing seamless control of the music whereever the user is currently located. Also, the user's movement around the environment can define the interaction with AirPlayer. When the user moves between zones, AirPlayer has the music follow the user, i.e. the music is switched on in the zone the user is moving to and switched off in the zone the user is moving from. In this way, AirPlayer supports interaction based on the proximity of users in the environment.

An example installation of AirPlayer is illustrated in Figure 4.1. The system is installed on a Mac Mini server PC located within the livingroom. Zones are connected wirelessly in each room of the house through Wifi via an AirPort Express device. The user can control the system from his smartphone from anywhere in the house directly through touch input or indirectly using the proxemic dimensions.

4.2 Interface Design

Besides the proxemic interaction principles, a direct input user interface was designed for AirPlayer. The server application is completely transparent on the user's computer, and all interaction happens trough the iTunes interface, allowing the user to manage their music collection, create playlists and so on. That is, no interface design is needed for the server application of AirPlayer.

The interface design for the mobile application of AirPlayer is a result of a series of iterative design sessions within the project group. The design was developed and refined during the sessions, and while the implemented design deviated somewhat from the session sketches, the sketches founded the grounds for the final interface design.

The mobile application consists of four underlying screen displays for the user to interact with, respectively the player, queue, music browser and panic screen displays. In this section, the interface design of these screen displays is described.

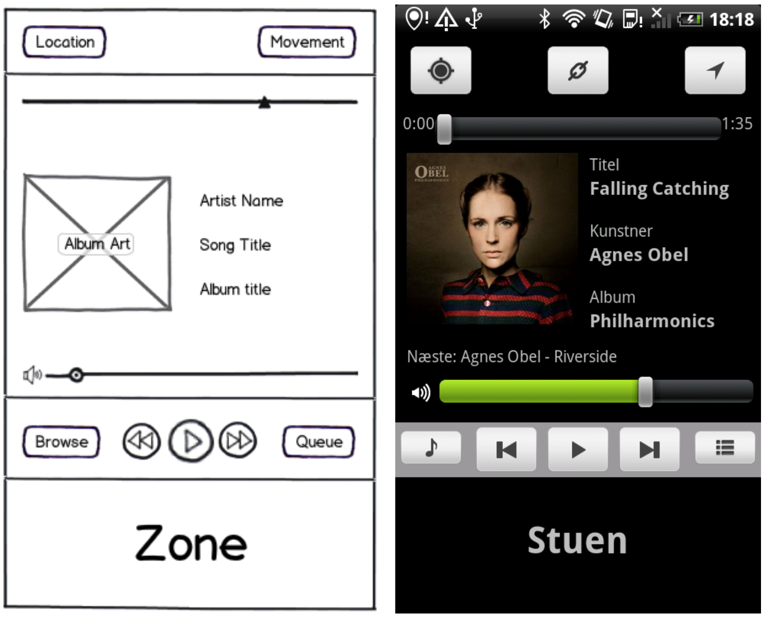
4.2.1 Player

The *Player* screen display is the main display of the application. This is the screen display the user sees when he first starts the application, and also this is the entry point for all the other screen displays in the application. The final sketch of interface design of the Player screen display as well as the implemented design is illustrated in Figure 4.2.

The Player screen display contains an interface for controlling the proxemic dimensions, an interface for controlling the playback of music, and an interface for controlling the zones in the system.

4.2.1.1 Proxemic Dimensions

The proxemic dimensions, i.e. the Location and Movement dimensions, are controlled from the top bar of the user interface as illustrated in Figure 4.3.



(a) (b)

Figure 4.2: (a) A sketch of the Player screen display. (b) A screenshot of the implemented design.

Figure 4.3: Control of proxemic dimensions.

Each of the proxemic dimensions are implemented as application modes which can be activated by the click on a button. When a mode is active, the button will be highlighted, and by pressing a highligted button the mode will be deactivated and the application will return to normal state. For example, if the user presses the Movement button, the Movement mode will be activated, and the music will follow the user around based on his movement around the environment. If the user presses the Movement button again, the Movement mode will be deactivated, and the music will continue its playback from where the user is currently positioned.

The interaction based on the proxemic dimensions is primarily based on the proximity within the environment, i.e. the interaction with the graphical user interface is only secondary. The user will activate one of the modes and probably use the application in the selected mode for a period of time. Because of the infrequent interaction with the graphical user interface, the controls for the proxemic dimensions are positioned at the top of the application interface.

Besides the controls to activate and deactivate the proxemic dimensions, the top bar also contains a control to activate a *Link* feature. While the link feature is not directly related to the proxemic dimensions, it is similar in that is a feature to control what music plays in which zones. When the user presses the link button, he is presented with a list of currently playing zones. From this list he is able to link the current zone to any of the listed zones. By linking a zone to another zone, the two zones will merge their queues and music stream, and start playback of the same music simultaneously. The user can link any number of zones together, for example to play the same playlist in all zones within the environment.

4.2.1.2 Music Playback

The music playback control is the dominating part of the user interface as illustrated in Figure 4.5.

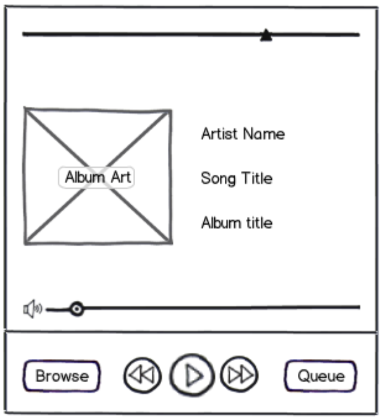


Figure 4.4: Control of music playback.

In the top of the control, the interface features a slider to seek to a specific position in the song progress. The music playback control also contains a large display of information regarding the music currently playing. The display shows albumart from the currently playing track, as well as artist name, song title and album title. Furthermore, the display shows information about the next song to be played, i.e. artist name and song title. Next, the control features a slider to adjust the volume, and lastly the control features a command bar for the user to start or stop the music, skip to next or previous track as well as opening the browse and queue activities. Again, the layout is structured such that the most frequently used functionalities are position closest to the user, i.e. in the bottom most part of the user interface.

4.2.1.3 Zones

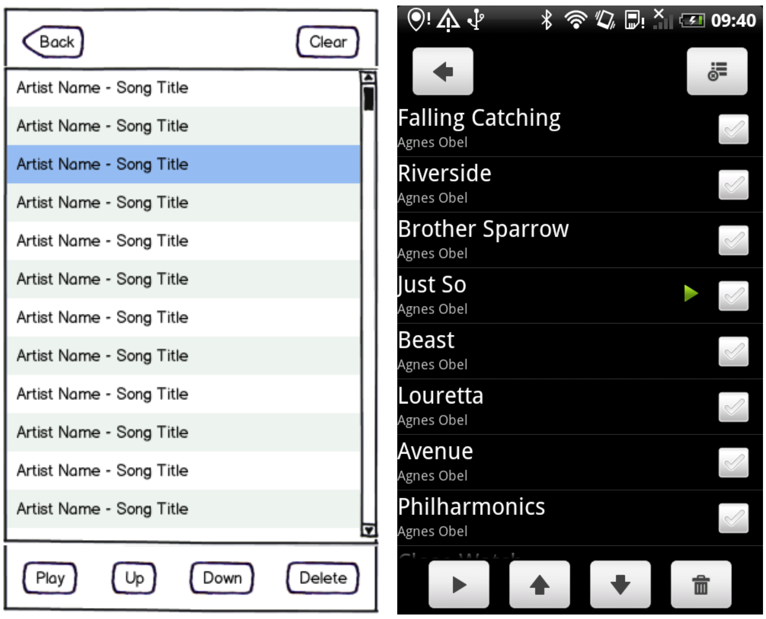
The zone control provides an interface for the user to control each individual zone in the system from the player screen display. The zone control is positioned closest to the user on the user interface, i.e. in the bottom of the application interface as illustrated in Figure 4.5.
The control shows the name of the current zone, and allows the user to switch to the other zones in the environment by using swipe gestures on the control, i.e. swipe left for the previous zone and swipe right for the next zone.

Figure 4.5: Control of zones playback.

4.2.2 Queue

The *Queue* screen display presents the current queue of songs for a particular zone as illustrated in Figure 4.6.

The queue screen display is accessed from the Queue button of the player screen display. The screen display features a top bar with a back button and a clear button. If the user presses the back button he is returned back to the player screen display. If the user presses the clear button, all currently queued songs in the current zone are dequeued and the music playback is stopped. The primary part of the queue screen display is a list of songs in the queue of a particular zone. The songs a listed by artist name and song title and the user can scroll in the queue by using swipe gestures. The bottom bar of the interface contains a play button as well as controls to manipulate the song queue. If the user marks a song on the list by touching it, and then presses the play button, the selected song will be played within the current zone. The up, down and delete buttons are used to manipulate the song queue. The up and down buttons moves the selected song up or down in the queue and the delete button removes a



(a) (b)

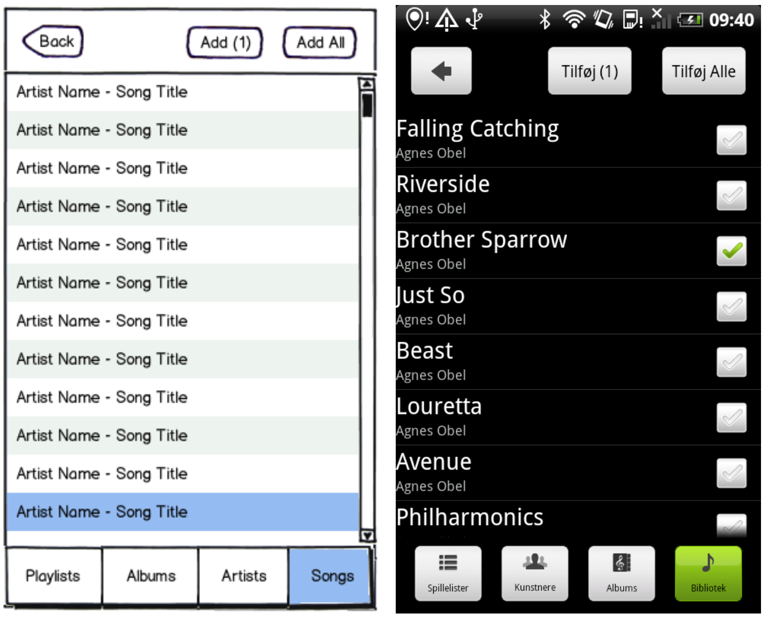
Figure 4.6: (a) A sketch of the Queue screen display. (b) A screenshot of the implemented design.

song from the queue.

4.2.3 Music Browser

The *Music Browser* screen display allows the user to browse the available music library. As described in the previous semester's project report [10], the music library is accessed through iTunes[3], where the user can organize his music, create playlists and so on. The music browser screen display is illustrated in Figure 4.7.

The music browser screen display is accessed from the Browse button of the player screen display. The screen display features a top bar with a back button, an add button and an add all button. If the user presses the back button he is returned back to the player screen display. If the user presses the add button, all selected songs are added to the queue of the current zone. If the add all button is pressed, all songs on the displayed list are added to the queue. The primary part of the music browser screen display is a list of items from the music library, i.e. playlists, albums, artists and songs. The items are sorted alphabetically and the user can scroll in the list by using swipe gestures. The bottom bar of the interface contains a tabbed interface for the user to browse through his music library. When the user selects a tab, the contents of the list is changed accordingly. For example, when the user selects the Albums tab, he is presented with a list of available albums from the iTunes library. From here, the user can select an album, and all songs on the album are displayed on the list. From the list, the user can select a number of songs to add to the zone queue, or simply add all songs from the album to the queue. In this way, the user can browse through his iTunes library from his smartphone, and add the songs he wishes to listen to to the queue of currently playing songs.



(a) (b)

Figure 4.7: (a) A sketch of the Music Browser screen display. (b) A screenshot of the implemented design.

4.3 Implementation

The implementation of AirPlayer is built upon a technical framework developed during the previous semester project. With a goal of deploying the application for testing in a real user environment, a solid implementation is required. During the master thesis, around 1100 man-hours have been put into developing the application, resulting in 9400 lines of source code divided between the server application and the mobile application respectively. With an application of this scale, many implementational details could be explored and explained in detail, however in this report only a selected set of significant features are covered.

First, implementational details regarding the data exchange within the system is described. The communication between the server and the mobile application progresses via a socket connection. Through the socket connection, information regarding the system is passed between the entities, e.g. information regarding the zones in the system, the music library and so on.

Lastly, the focus for this master thesis is proxemic interaction in a multi-room music system, thus the implementation of the proxemic dimensions is a key aspect to be described.

In this section implementational details regarding these key functionalities of AirPlayer is covered. Through the section, the following technical terms are used to describe the implementation:

- **Application** defines the overall mobile application, in this case AirPlayer. The application contains a set of classes, services, activities and views, together comprising the application.
- **Activity** defines a single executable operation or screen display that the user can interact with. An activity is responsible for creating a user interface and contains all underlaying business logic.
- **Service** defines an application component that is able to perform long-running operations without the need for user interaction, i.e. an application component that runs in the background.
- **Listener** defines an application component that listens for events within the application. An event can be triggered from for example a service when a certain state of operation is reached.

4.3.1 Socket Communication

To exchange data between the mobile application and the server application, a socket connection is established. The socket connection is based on the TCP protocol using a client-server request-response communication protocol, with the server application as a central server and the mobile application as a client. Through the socket connection, the server is capable of serving multiple clients. The socket connection is illustrated in Figure 4.8.

When the mobile application starts it sends a TCP `SYN` request to the server IP address to setup a new socket connection. The server responses with a TCP `ACK` message, and the connection is

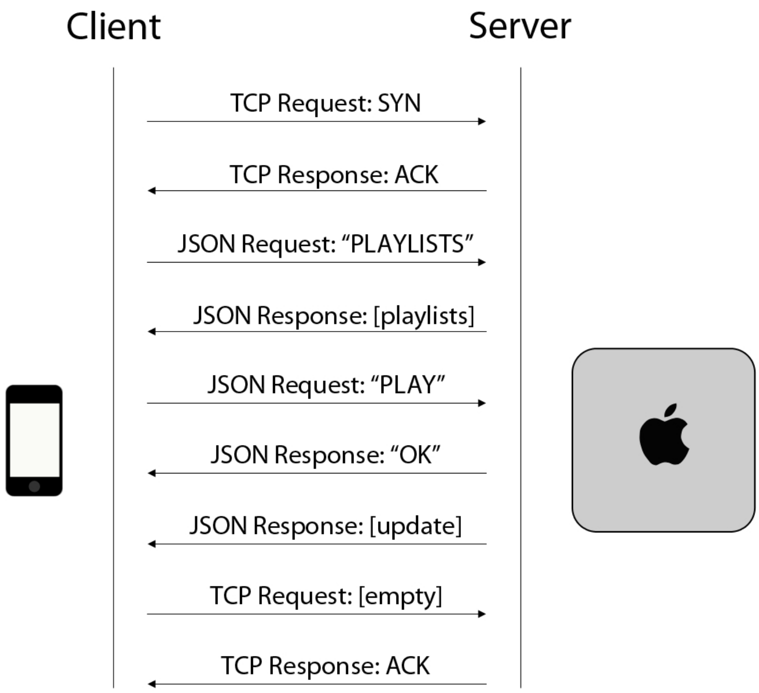


Figure 4.8: Data exchange between client and server through TCP socket.

established. Now the connection is open, and messages can be passed between the server and the client and vice versa, e.g. the client can request a list of music albums and the server will respond accordingly, returning a list of available albums from the music library, or the client can send a play command and the server will start the music in the selected zone accordingly. The exchanged data is JSON encoded and parsed on both the client and server applications. When the client disconnects, an empty message is sent to the server, and the server returns a TCP `ACK` response and removes the client from the list of connected clients.

An extract of the implementation of the server side of the socket communication is shown in Source Code 4.1. The `ClientService` class runs as a background service with an infinite while loop listening for connecting clients on lines 4-9. On line 6 a new `clientSocket` connection is established using the `AcceptTcpClient` blocking method of the open socket. On line 7 the client connection is added to a list of open connections and on line 8 a new socket reader is started with the `readSocket` method to read incoming data from the connected client.

When a new client is connected, the service starts a new reader to read incoming messages from the client. An extract of the reader method is shown in Source Code 4.2. On lines 5 the client stream is stored in a variable and on line 6 a message buffer variable is instantiated. On line 7 a byte buffer is instantiated to contain the incoming messages. On lines 9-29 an infinite while loop is started to listen for incoming messages. On line 11 the `Read` method blocks for incoming messages, and when a message is received, the message is stored in the buffer. On line 13 a check is performed to see if the length of the message is greater than 0. If not, the socket connection has been closed, and the client is removed from the list of connected clients on line 26. If the length of the message is greater than 0, the message is encoded as a

```
public class ClientService
{
   ...
   while (true)
   {
      clientSocket = new Client(serverSocket.AcceptTcpClient());
      ClientHandler.Instance.Clients.Add(clientSocket);
      readSocket(clientSocket);
   }
}
```

Source Code 4.1: The `ClientService` class on the server application is listening for connecting clients in an infinite `while` loop.

string and added to the message buffer on line 15. Lines 17-22 iterates over the messagebuffer and extracts the individual messages seperated by a linebreak ("\n"). When a single message is extracted, the `messageReceived` method is called to parse the message on line 21.
When a message is received from the client, is it parsed to extract the message information and perform the required actions accordingly. The message parsing is handled in the `MessageReceived` method as shown in Source Code 4.3. The method reads the message content and performs a switch to perform the required actions on lines 4-11. Line 5-8 defines the actions required when a `PLAY` message is received from a client. Similar cases are implemented for other functionalities in the application.

Similar functionality have been implemented on the client side to parse data received from the server, thus the server application can exchange messages with the clients through the socket connection and vice versa. In this way, the socket connection is used to handle all requests within the system, i.e. controlling the music playback, adjusting the volume, browsing music, editing playlists and so on.

4.3.2 Proxemic Dimensions

The implementation of proxemic dimensions is a central part of the application and the focus for this master thesis. The proxemic dimensions are depending on the users current position within the digital environment, i.e. the users position relatively to the active zones in the system.

The users position is tracked using a simple implementation of Wifi triangulation through the mobile application. For the prototype application, only coarse grained information regarding the users position is required, i.e. in which zone the user is currently positioned and whether his position has changed over time. This means that the users exact coordinates are unimportant, and thus a simple implementation is sufficient. By using results from the Wifi signal strength of the AirPort Express devices, each representing a zone in the system, the relative position of the user can be approximated, thus creating the basis for implementing the proxemic dimensions. As illustrated in Figure 4.9, the users position is approximated based on the current signal strength of each available Wifi access point, calculated on the users smartphone.

```
private void readSocket(Client client)
{
    Thread thread = new Thread(() =>
    {
        NetworkStream stream = client.GetStream();
        String messagebuffer = "";
        byte[] buffer = new byte[BUFFER_SIZE];

        while (true)
        {
            int read = stream.Read(buffer, 0, BUFFER_SIZE);

            if (read > 0)
            {
                messagebuffer += Encoding.UTF8.GetString(buffer, 0, read);

                while ((int i = messagebuffer.IndexOf("n")) > -1)
                {
                    string msg = messagebuffer.Substring(0, i);
                    messagebuffer = messagebuffer.Remove(0, i + 1);
                    this.MessageReceived(msg, client);
                }
            }
            else
            {
                ClientHandler.Instance.Clients.Remove(client);
                break;
            }
        }
        client.Close();
    });
    thread.Start();
}
```

Source Code 4.2: The `readSocket` method starts a new thread to listen for incoming messages from connected clients on the open socket connection.

```
public void MessageReceived(String message, Client client)
{
    ...
    switch (message)
    {
        case "PLAY":
            zone.Play(...);
            break;
        default:
            break;
    }
}
```

Source Code 4.3: The `MessageReceived` method parses the messages to extract the message information and perform the required actions.

The access point with the highest signal strength is the closest, and thus represents the zone that the user is currently positioned within. The access point with the next to highest signal strength is the second closest and so on, and if the ratio changes over time, it indicates that the user is moving. In the illustrated example, the mobile application receives scan results from 3 different access points, i.e. kitchen, bathroom and livingroom. Based on the signal strength, the kitchen access point is approximated to be the closest zone with 80% signal strength, while living room is the second closest with 15% and bathroom has the longest distance with a signal strength of only 5%.

To use results from the Wifi signal, AirPlayer utilizes the `WifiManager` class from the Android standard library. The class contains methods for scanning available networks, calculating signal level for available access points, getting info regarding the current Wifi connection and so on. The `WifiManager` interface is wrapped within the `WifiService` service class as shown in Source Code 4.4. By implementing the `WifiManager` in a service, it will run in the background and be available throughout the entire lifetime of the application. In this way, the service and the provided results can be accessed from anywhere in the application where needed.

```
public class WifiService extends Service
{
   ...
   private void Scan()
   {
      List<WifiScanResult> results = new ArrayList<WifiScanResult>();

      connectivityManager = (ConnectivityManager) ↵
          getSystemService(CONNECTIVITY_SERVICE);
      networkInfo = ↵
          connectivityManager.getNetworkInfo(ConnectivityManager.TYPE_WIFI);

      wifiManager = (WifiManager)getSystemService(Context.WIFI_SERVICE);
```

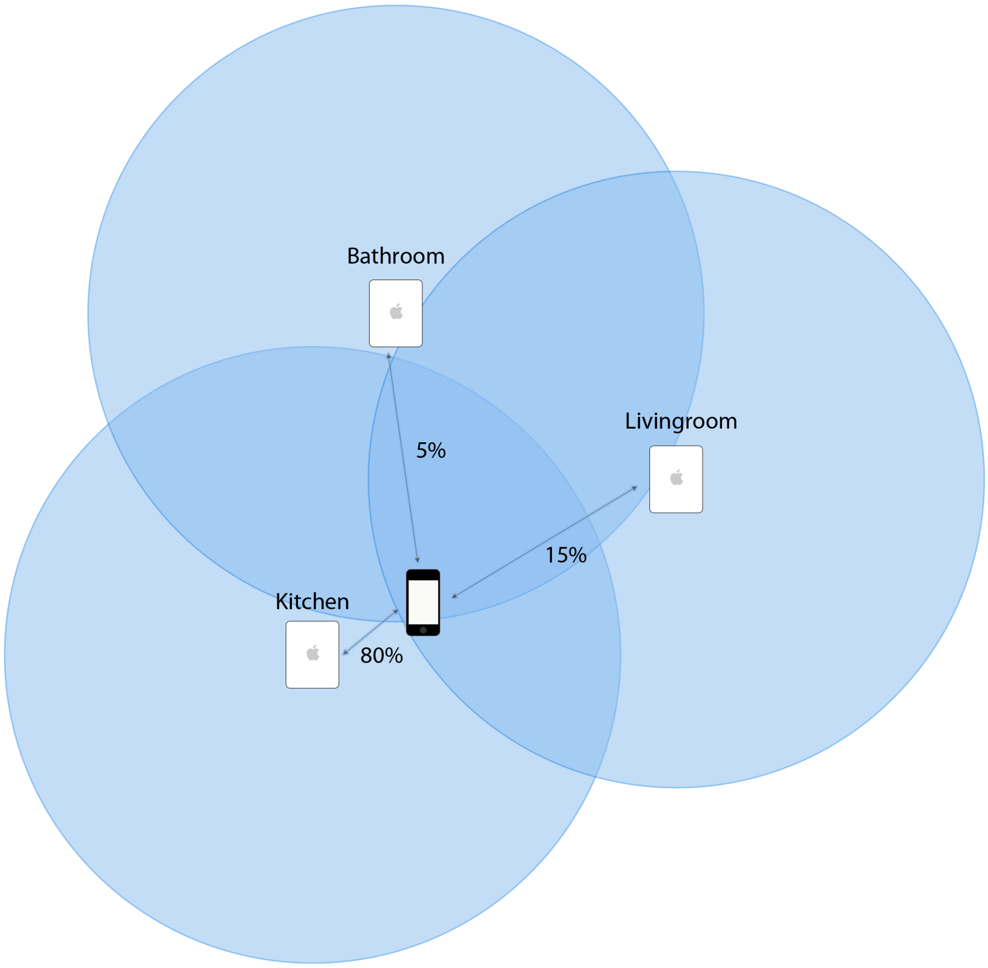


Figure 4.9: The users position is approximated using a simple implementation of Wifi triangulation.

```
        wifiInfo = wifiManager.getConnectionInfo();

        myWifiManager.startScan();

        if (networkInfo.isConnected())
        {
            for (ScanResult result : wifiManager.getScanResults())
            {
                if (result.SSID.equals(wifiInfo.getSSID()))
                {
                    results.add(result));
                }
            }
        }
        return results;
    }
```

```
28
29     public Zone GetClosestZone()
30     {
31        zonehandler = ZoneHandler.getInstance();
32        List<Zone> zones = zonehandler.getZones();
33        List<WifiScanResult> wifiResults = this.Scan();
34
35        Zone closest = null;
36
37        for (WifiScanResult result : wifiResults)
38        {
39           if (zones.contains(new Zone(result)))
40           {
41              if (closestresult == null)
42              {
43                 closest = result;
44              }
45              else if(result.strength > closest.strength)
46              {
47                 closest = result;
48              }
49           }
50        }
51        return new Zone(closest);
52     }
53     ...
54  }
```

Source Code 4.4: The `WifiService` service class with the `Scan` method returning results from all available Wifi access points, and the `GetClosestZone` method, returning the closest zone based on the scan results.

Source Code 4.4 shows an extract of the implementation of `WifiService`. The service contains a private method, `Scan` for scanning for available networks, and a public method, `GetClosestZone` which returns the currently closest zone based on the scan results. The `Scan` method is defined on lines 4-27. On line 6 a new `ArrayList` is instantiated to contain the results from the Wifi scan. On lines 8-9 a `ConnectivityManager` is instantiated to retrieve information regarding the current network connection. On lines 11-12 a `WifiManager` is instantiated to get information about available Wifi accesspoints, and on line 14 the scan is started. On line 16 a check is performed to ensure that an active network connection is available. Lines 18-24 iterates over the received scan results. On line 20 a check is performed to filter out any results that are not associated with the currently connected Wifi network, i.e. results from any access point that is not a part of the currently connected Wifi network is not available as a zone in the system, and thus is filtered out. On line 22, the result is added to the list of results, and on line 26 the list is returned to the caller. The `GetClosestZone` method is defined on lines 29-52. On line 31 a `ZoneHandler` is instantiated. The `ZoneHandler` contains information regarding available zones in the system. From the `ZoneHandler` the list of available zones is retrieved on line 34. On line 33 a scan is started and the result is stored in a list. Lines 37-50 iterates over the results from the scan to find the zone with the highest signal strength level, i.e. the zone which the user is currently positioned within. The check on line 39 ensures that only results which are related to active zones within the system are considered. On line 51 the closest zone is returned to the caller.

To use the `WifiService` from within an activity, the activity will have to bind to the service and register itself as a listener for events on the service. Source Code 4.5 shows an extract of the implementation of the `SocketActivity` activity class. `SocketActivity` functions as a base activity for all other activities in the application, and defines common properties and functionalities. On line 5-12 the `onStart` method is defined which is called whenever an activity gets in focus in the application. On lines 10-11 the activity binds to the `WifiService` through a the defined `wifiintent`. When a service connection is established, the activity is automatically registered within the service as a listener. When registered as a listener, the activity will automatically receive updates when the service triggers an event, e.g. when the user moves to a different zone and the signal strength changes.

```
1  public class SocketActivity extends Activity
2  {
3     ...
4     @Override
5     public void onStart()
6     {
7        super.onStart();
8
9        // Bind to WifiService
10       Intent wifiintent = new Intent(this, WifiService.class);
11       bindService(wifiintent, connection, Context.BIND_AUTO_CREATE);
12    }
13    ...
14 }
```

Source Code 4.5: The `SocketActivity` base class binds to the `WifiService` when started.

With this simple implementation of Wifi triangulation, the application is ready to implement the proxemic dimensions.

4.3.2.1 Location

The location dimension is based on the users current location relatively to other entities in the digital ecology. This dimension requires awareness of the users current posibion within the environment. The location dimension is implemented as a mode that can be activated in the `Player` activity, i.e. the activity that displays the player interface to the user. When the user activates the *Location* mode, the application automatically changes the user interface as the user changes position in the environment, thereby assisting him to always control the zone which he is currently located in.

The Player activity is bound to the WifiService and uses the received Wifi results to estimate the users current location within the environment. Based on the current Wifi strength from the available access points, the users current location is estimated, as illustrated in Figure 4.9 on page 33.

The signal strength from the access point in the kitchen zone is currently the strongest, and thus the user is estimated to be located within the kitchen zone. If the users location changes, i.e. he is moving into another zone, the interface will change to reflect the change in the users location, and allow the user to control the music in the new zone.

The implementation of the location dimension is examplified in Source Code 4.6. The PlayerActivity is bound to the WifiService and whenever a result is received from the service, the WifiResultsReceived method is triggered on lines 3-19. WifiResultsReceived checks if the application is currently in Location mode on line 6. On lines 8-17 the application iterates over the received results and compares the signal strength to the users current location on line 12. If any of the received results has a signal strength higher than the saved signal strength from the users current location, the user has moved to a different zone and thus the change of location is reflected on the interface on line 14.

```
1  public class PlayerActivity extends Activity {
2     ...
3     public void WifiResultsReceived(List<WifiScanResult> results)
4     {
5        ...
6        if (locationmode)
7        {
8           for (int i = 0; i < WifiService.getScanResults().count(); i++)
9           {
10             Zone newzone = ZoneHandler.getZone(i);
11
12             if(newzone.signalStrength > oldzone.signalStrength)
13             {
14                flipView.setDisplayedChild(newzone);
15                break;
16             }
17          }
18       }
19    }
20 }
```

Source Code 4.6: The PlayerActivity class with the WifiResultsReceived method which changes the interface to reflect the users current location.

4.3.2.2 Movement

The movement dimension is based on the users movement around the environment. As with the location dimension, the movement dimension requires awareness of the users current position within the environment, but also knowledge about changes in the users position over time. The movement dimension is also implemented as a mode that can be activated in the `Player` activity. When the user activates the *Movement* mode, the application plays the current music stream whereever the user is located in the environment. If the user moves to a new zone, the music follows him around, thereby ensuring a continous music experience.

To implement the movement dimension, the `Player` activity is bound both to the `WifiService` and the `SocketService`, and sends the received Wifi results to the server via the socket as the user moves around the environment. Based on the change of Wifi signal strengths from the available access points over time, the users movement is estimated, as illustrated in Figure 4.9 on page 33.

The signal strength from the access point in the kitchen is currently the strongest, and thus the user is estimated to be located within the kitchen zone. If the user moves to a new zone, the server application will receive an update through the socket connection, and based on the received Wifi results, the music will be started in the new zone and turned off in the old.

The implementation of the movement dimension is examplified in Source Code 4.7. When the movement mode is activated, all zones are merged into a single zone object, to enable synchronous playback of the same music throughout all the zones. When the server application receives the Wifi scan results through the socket connection, they are passed forward to the zone object which is then responsible for turning the music on or off in the respective zones. Source Code 4.7 shows an excerpt of the implementation of the `Zone` class of the server application. Whenever the server receives the Wifi scan results through the socket, the `ParseMovementResults` method is invoked as defined in lines 4-22. The method iterates over the received Wifi results on lines 7-21. For each result, a ratio between the signal strength of the currently closest zone and the scan result is calculated on line 11. This ratio is used to turn the music on or off in the respective zones. Whenever the ratio is above 85%, i.e. the signal strength of the new zone is 85% of the signal strength of the current zone, the music is turned on in the new zone on line 15. Similarly, if the ratio is below 72%, the music is turned of on line 19. The ratio boundaries were based on a set of experiments in different test setups, and the boundaries were adjusted until a suitable ratio was found. By using assymetric boundaries, situations are avoided where the system continously turns the music on and off when the user is located around the boundary between two zones.

4.4 Limitations

While AirPlayer supports the required functionality to deploy the prototype multi-room music system in a field study, some limitations are enforced to the system.

4.4.1 Three Zones at Maximum

In theory, the implementation enforces no limitations to the number of zones in the system, however the utilized protocol creates an upper boundary for the amount of zones that can play simultaneously. AirPlayer streams music through the AirPlay protocol via a Wifi-network to a set of connected AirPort Express devices and the system is remote controlled from a HTC Desire smartphone via the same Wifi-network.

```
public class Zone
{
  ...
  public void ParseMovementResults(List<WifiScanResult> scanresults)
  {
    ...
    foreach (WifiScanResult scanresult in scanresults)
    {
      Zone newZone = new Zone(scanresult);

      double pct = (double)((double)currentZone.signalStrength / ↵
         (double)newZone.signalStrength);

      if (pct >= 0.85)
      {
        newZone.Start();
      }
      else if (pct < 0.72)
      {
        newZone.Stop();
      }
    }
  }
}
```

Source Code 4.7: The `Zone` class with the `ParseMovementResults` method which changes the music in the individual zones as the user moves around the environment.

While the AirPort Express devices supports the 802.11n standard for Wifi-networks allowing for a theoretical maximum data transfer speed of 600Mb/s, the HTC Desire supports only the 802.11g standard allowing for a maximum data transfer speed of 54Mb/s. Connecting a 802.11g device to a 802.11n network results in all devices to be degraded to the lower speed of the two to ensure consistent communication. This means that the network effectly supports a maximum theoretical data transfer speed of 54Mb/s.

The audio being streamed is encoded into RAW audio at a sample rate of 44.1kHz with 16 bits per sample in dual channel to support stereo. This results in a data stream of 170kB/s from each zone. Since each of the connected AirPort Express devices is configured to extend the Wifi network, all traffic in the network is routed through every single access point, resulting in exponential data load with each zone in the system.

This results in a theoretical maximum of 6 concurrent zones in the system. The theoretical maximum is only reachable under optimal conditions in a clinical environment however, and through stress testing of the system is was found that 3 concurrent zones is the maximum amount of zones to ensure stable performance of the system in the current configuration.

4.4.2 Short Distance

As with the number of concurrent zones, the implementation enforces no theoretical limitation to the distance between the entities in the system. Entities within the system are connected through the WiFi network, and thus the entities obviously need to be within Wifi range of each other.

The connection through Wifi poses some other challenges related to the interaction through the smartphone interface however. The hardware implementation of the HCI Desire does not fully comply with the 802.11 standards, and thus does not allow for WiFi roaming. This means, that when the smartphone is connected to a Wifi access point in one zone, it does not switch connection to the access point in any other zone, even though one with a significantly stronger signal might be available. Instead it waits until the signal to the current access point is lost, whereupon it starts a scan for access points within range and reconnects to the access point in the current zone if available. In this way the smartphone temporarely looses the Wifi connection when the current access point gets out of range, and while the proxemic dimensions rely entirely on results from the Wifi signal strength, a continious Wifi connection is required. To accomodate for this, AirPlayer should be installed so that each entity can connect directly to any other entity within the system. In this way, no matter where in the system the smartphone is located, it can be connected to the same Wifi access point and still be within range of all access points in any zone in the system.

4.4.3 Only One Zone During Movement

AirPlayer streams music to a set of connected AirPort Express devices each representing a zone in the system. The AirPort Express device then plays the music through the attached speakers. Each AirPort Express device contains a data buffer for buffering the audio packets when received. Since AirPlay is a proprietary closed protocol and there is no available documentation, it is not possible to access or modify this buffer. The size of the buffer can be estimated, though in practice there is no way of knowing the exact amount of buffered data on each of the AirPort Express devices.

When playing the same stream of music in multiple zones, it is important that the zones are synchronized. To ensure this, any action related to merging or separating zones results in a short pause to clear the buffer on each device and restart the music stream.

These small pauses are undesirable however when the user is using the movement dimension of the proxemic interaction principles. For the zones to stay in sync when the user moves between zones, the music would be interrupted whenever a zone starts or stops playing the music stream, and this would ruin the music experience. To avoid these breaks in the music, and to keep the music in sync even while moving between zones, all zones are merged into a single group of zones, each playing from the same music streams at all times. To create the illusion of the music following the user around as he moves between zones, the volume is simply turned to a minimum whenever the user is leaving a zone, and turned back to the currently configured volume when the users enters a zone. This also means that whenever the user is utilizing the movement feature, only one zone is available within the system.

4.4.4 Single User

Introducing multiple users to interact with the same system simultanously poses several new design challenges. For example, what happens if multiple users control the music in the same zone, thus interfering the music experience of other users? Should it be possible to gain exclusive control over a single zone? When movement is enabled, how should presence of multiple users in the same zone be handled?

Several similar questions could be taken into account, however, while aspects regarding multiple users within the system is a relevant concern, it is out of scope for this master thesis to take these aspects into consideration. Therefore, AirPlayer is limited to serving only one user at a time.

4.5 Summary

In this chapter, the entire AirPlayer system was described. AirPlayer is a prototype multi-room music system with support for the proxemic principles in the interactions design.

Besides the proxemic interaction, the system features a direct input interface for touch input. Through the touch interface users can control the music playback within the system, handle music within each zone as well as activate the proxemic interaction interfaces as system modes.

Data is exchanged between the server application and the client application through a TCP socket connection using JSON encoded strings. Through the socket, the client can send commands to the server, e.g. play and pause the music, adjust the volume or add songs to the queue. The proxemic dimensions, i.e. Location and Movement, are implemented using a simple implementation of Wifi triangulation. Based on the signal of available Wifi access points in the environment, the users position is approximated. The position can be used to track the users location, i.e. the currently closest zone relative to the user, or to track the users movement, i.e. track the change in the users position over time.

Due to timing constraints of the master thesis as well as shortcomings in the selected platform, certain limitations are enforced in the system. The system can handle up to three zones at most, placed within short distance of one another. Furthermore, the system handles all zones as one linked zone when the movement mode is activated. Lastly, the system is limited to handling only one user at a time.

CHAPTER 5

FIELD STUDY

To evaluate the integration of the proxemic dimensions in the interaction design of AirPlayer, the system was deployed for testing in an in-situ field study. AirPlayer was installed in the homes of a selected group of test participants, and integrated as a part of the participants' existing multi-room music systems.

By testing the system in a field-study the test participants are able to integrate the system in their daily routines in the same manner they use their current multi-room music systems. In this way, the proxemic dimensions can be studied in a realistic environment, thereby providing a better understanding of the impact of the interaction design in practice.

Through the test period, the test participants logged their experiences and thoughts in a log and the test period was concluded with a closing interview. Also, AirPlayer logged all events in the system. After the test period, the collected data was analyzed to identify the effect of the implemented proxemic dimensions in the interaction design of AirPlayer. In this chapter, the applied method for the field study is documented. Furthermore, the collected data is analysed and evaluated upon.

5.1 Method

The field study was performed with the goal of gathering empirical data regarding the effect of integrating the proxemic dimensions in the interaction design of a multi-room music system.

This section documents the method utilized for the conducted field study.

5.1.1 Participants

Two families with a total of three family members participated in the conducted field study. Both families already owned a multi-room music system, and thus were familiar with the basic concepts of a multi-room music system.

The families were denoted A and B for the purpose of reference, and the individual family members denoted as A1, A2 and B1 for the families A and B respectively. The participants were 16, 28 and 47 years of age, and had between one and five years of experience with their current multi-room music systems. The participants received a small token of appreciation for participating in the field study of AirPlayer.

5.1.2 Setting

The field study was conducted at the respective homes by the participating test families. This setting provided a familiar and natural environment for the test participants to explore the new interaction design of their multi-room music system.

Similarly, the concluding interview with the test participants was conducted in the homes. Two different roles were assumed during the interviews; the interviewer and the observer role. The responsibility of the interviewer was to conduct the actual interview whereas the observer would stay in the background and observe, thereby not participating directly in the interview. This ensures that one person was allocated to following the semi-structured interview while the other would note various observations during the session.

5.1.3 Procedure

The field study was conducted at home by the test participants. The test period spanned over three weeks, where AirPlayer was installed and integrated with the users existing multi-room music systems. The participants were introduced to the system, and afterwards the users would use the system in their everyday routines throughout the test period.

At the end of the test period, a concluding interview was conducted with each of the participating families. The interviews were structured as semi-structured interviews following the guidelines of [16]. The semi-structured approach provided more open-ended interviews, and left room for clarification and following comments from the participants. During the interviews, a set of different topics were covered with the purpose of exploring various areas of interest within the problem domain. The interview guide covered the following areas:

Formalia: The interviews were started with formal information about the study, and how the interview would progress.

Generally: The first topic of the actual interview covered general aspects regarding the users experience with AirPlayer as a multi-room music system.

Location: Afterwards, the interview guide suggested a series of question specifically targetting the users experience of the implementation with the Location dimension in the interaction design.

Movement: Lastly, the interview contained question specifically targetting the implementation of the Movement dimension in the interaction design.

The interview guide (Danish) is located in Appendix A on page 67.

5.1.4 Data Collection

During the test period, the test participants noted their thoughts of the system in a log. The log would then function as a basis for a conversation regarding the system in a concluding interview. The interviews were recorded using a voice recorder on a mobile phone, and were later transcribed. Furthermore, the observartions obtained by the observer were documented in handwritten notes, containing information regarding esoteric remarks, visual references etc. during the interview.

The transcriptions of the performed interviews (Danish) is located in Appendix B on page 71 and Appendix C on page 79.

Furthermore, data was logged through an automatic logging functionality implemented in AirPlayer. The system logged all events occuring in the system, and recorded all interaction points between the users and the system.

5.1.5 Data Analysis

The collected data was analyzed jointly by both authors of this master thesis. After the test period, the transcriptions of the interviews, as well as the notes taken by the observer during the interviews were reviewed. Furthermore, the automatically generated log files were queried to identify areas of interest regarding the users usage of the AirPlayer, in particular log entries related to the users interaction with the system.

5.1.6 System Modifications

While AirPlayer is a functional prototype, and hundreds of man-hours have been put into building a solid multi-room music system, unpredicted errors can still occur when deploying the application in a real user environment. Therefore, a few modifications were made to the system.

A panic button was implemented on the mobile remote application, as illustrated in Figure 5.1. When the button is pressed, the application is restarted from scratch. The panic activity is not intended to be included in a final release version af the application, it was implemented only as an easy way for the test participants to restart the application if anything goes wrong during the test period. The application is still a proof-of-concept prototype, and some errors can occur in specific scenarios. To avoid having the test to stop as a result of sporadic errors, this feature was implemented as a quick way for the test participants to start over with using the application with only limited inconvenience and without involving the developers. The *Panic* screen display is accessed by pressing the physical menu button on the Android smartphone from any other screen display in the application.

(a) (b)

Figure 5.1: (a) A sketch of the Panic screen display. (b) A screenshot of the implemented design.

Also, a logging functionality was implemented in the system, to automatically log events and interaction points. The log files provided essential data for the data analysis, as presented in the following section.

5.2 Findings

This section covers the findings obtained through an analysis of the conducted interviews and log files. Firstly, the findings of the interviews from both households are presented, covering both general system aspects as well as specific findings related to the two proxemic dimensions, Location and Movement. Afterwards, an analysis of the log files are presented, containing quantitative data related to the use of the system in the test period of the system.

5.2.1 Qualitative Data Analysis

In order to obtain an understanding of how the two proxemic dimensions are received amongst the users of a multi-room music system, we had to ensure that AirPlayer had the features which the users were accustomed to. That is, using AirPlayer was not to be perceived as using an inferior system in terms of basic controls and functionality compared to what the users are used to.

Both families gave the impression that they were satisfied with the basic functionality of AirPlayer, and both families considered AirPlayer to be very similar to the systems that they already had in their household. An implementational detail was however noted by family A in which they desired an option to enable or disable a repeat function in AirPlayer.

> *The idea behind the system is very nice, but I lack a few finesses. We had queued Kim Larsen in the system, but after two days we discovered that we had been listening to the same over and over again, as it repeats itself. I thought it stopped when it was done playing the music queue.* (A1)

As the user expresses, she expected the system to stop playing when it had finished playing the music queue, however the implementation of this was intentional as it served the purpose of keeping the music system playing, which was ideal for testing purposes. In retrospectrum, if AirPlayer had an implementation of a repeat function, it would be more similar to what the users were accustomed to, and should therefore had been implemented.

Additionally, the users expressed a desire for a larger queue size, as they experienced that the queue would become full after adding their songs to the given queues in the given zones.

> *..I experienced that the song queue would become full - even after I had listened to the music* (B1)

This is a direct results of both technical limitations, as covered in Section 4.4, but also because the songs were not removed after having being played. Had either the repeat function been implemented such that played songs would get removed after having being played, or if the technical foundation supported a longer list of songs, this could easily have been accomodated. The user also noted that this was not a real problem as he was able to edit the queue from the remote application.

In terms of accomodating problems during the test period, both households expressed that they were able to solve issues by using the implemented panic button on either the server- or the client application.

> *If we ran into any problems, we just used the panic button [in the remote application], and then it worked.* (A2)

As stated, household A experienced that they were able to resolve any problem they ran into by using the panic button on the remote application. This provided the households with an immediate way to solve minor issues in AirPlayer. During the interview with household B, when asked whether or not they experienced any problems when using the Location function, they answered the following:

> *No, not according to what I can remember off-hand. Well we did once experience that the [system] behaved strangely after I edited my iTunes library. It was related to playlists and stuff. But after I restarted the system [using the panic button on the server application], everything worked fine again.* (B1)

This shows that both the panic button on the remote application and the server application was able to resolve issues without having to interfere with their testing. In relation to the panic button, an interesting discovery where made, as the users from household A expressed the desire of a similar functionality in their existing system, as they were asked what was missing in AirPlayer compared to their current setup:

> *I would like that we had the panic button in Sonos. It would like to have to use it the other way around, but that was not what you asked me about. It was nice to have the panic button which enabled me to start over though.* (A1)

This shows a desire to be able to reset the state of the entire music system by using the functionality of the panic button. Data from the log file supports this observation, where it can be seen that the panic button on the Android application had been used multiple times throughout the test period even though the system was not reporting any errors.

5.2.1.1 Location

An essential part of this master thesis was to uncover how the users perceived and received the two proxemic dimensions, Location and Movement. As a part of the interviews performed, we asked a series of questions directly related to these dimensions, as well as utilized the log files to see how both functionalities were used and adopted by both families. This section covers the findings of the Location functionality.

Both families considered the Location dimension to be intuitive, but also very transparent. In fact, the feature worked so well that the user A1 was not even aware that she was using the feature:

> *To me it had to be fantastic, because I used it and I didnt even think about whether or not the feature was enabled. I actually thought it was connected with [the Movement function]. It worked when I adjusted the volume and other things. But I had not thought about the fact that I used it, but I really did.* (A1)

This statement showed that the feature was well integrated in AirPlayer, but also very subtle and simple. The fact that she was not aware of having enabled the feature despite of not having it in her own Sonos system also shows that the feature was very intuitive to use.

Similar findings were found in household B, where they found the function to be simple and useful. Compared to household A, they were very much aware of this feature, as they realized that having this functionality in their system enabled them to control the zone in which they were located:

> *As I said earlier, I think that a relatively simple function as this one is extremely good. The fact that I do not have to find the room that I am about to play music in, that makes it easy to utilize the mobile phone to control the music according to your current location.* (B1)

By enabling the Location function, the participant found it easy to use the remote application to control AirPlayer, as the application was aware of its current location and could therefore easily manage the music in the given zone.

Both families used the Location function during the majority of the test period. As the participant from household B expresses, he had no reason not to use it as he did not need to control anything else than the zone he was in:

> *I would estimated that I just used it.. I just used it most of the time. Actually, I have had no real need to control another room other than the one I was present in.* (B1)

Which is also supported by the previous comments from *(A1)* who utilized the Location functionality without being aware of it, as she experienced that the system responded to her interactions in the zone in which she were.

The test participants were also asked to evaluate the usefulness of both the Location and the Movement functionality in AirPlayer. Both families were positive towards the functionality, which is expressed by the participant from household B:

> *The fact that I do not need to navigate to the room that I am about to play music in, well that makes it very easy to use the mobile phone to control the music where you are.* (B1)

This opinion regarding the usefulness of the Location functionaility is also shared by household A. The log files from both families also reveal that the swipe functionality used to select specific zones in the households were rarely used, as the Location fuction was enabled most of the time.

5.2.1.2 Movement

The second proxemic dimension, Movement, was also implemented as a part of the interaction design of AirPlayer. This section covers the findings directly related to the Movement functionality, where both findings from the performed interviews as well as the results from the log files are used.

Both families found that the Movement function worked well, and differed from their regular music setups in the way they interact with their music system. As one of the participants from household B said:

> *We have used it a lot, partly because we wanted to test the system, but also because we found it to be clever and fun to use. The thing about having the music following you is nice when you walk around at home in your own thoughts.* (B1)

Household A shared this opinion towards using the function, however they experienced slightly unintended behavior during the test period. When asking how the function behaved, they said that the music should overlap for longer periods than it currently did:

> *It could have waited 15 seconds before it stopped the room you were leaving, and then start the music immediately as you walk into the next room. I think more overlap would have worked wonders* (A1)

As participant A1 said, she would have liked AirPlayer to have a timer based threshold instead of having one purely based upon the distance between two zones and the remote control. The reason for this behavior that she experienced was most likely due to the layout of their house and the placement of the AirPort Express devices. Household B did not experience similar behavior when using the Location functionality.

At times, the participants experienced that the music would not be entirely synchronized, which were obvious when standing in the middle of two different zones while using Movement. The result was that the participants experienced a slightly asynchronious playback of the same song:

> *As I said earlier, there were a few incidents where the music was not entirely synchonized, but we learned that we could fix it by changing the track being played, and then switching back* (B1)

The asynchonious behavior was caused by technical limitations discovered in the preliminary study when the technical framework was developed. As B1 states, he found out that this could be solved by forcing the system to reload the songs, and thereby getting the music back in sync.

Both families expressed no doubts regarding the use of the functionality. When asked to describe the function using their own words, they were able to give a concise description of how the function worked:

> *I think that I quickly obtained an understanding of how it worked. I mean, it just starts the music before you enter the next room, and then turns the music off when you leave. It is nice and easy.* (B1)

As B1 says, he finds the function easy to use as the purpose of the function is clear to the user. But as he also elaborated:

> *Of course, I had to get used to how it worked the first time I used it, but I do not think that we had any doubts of how it was supposed to work.* (B1)

Household A also found the Movement function easy to use, they explained the purpose of the function as follows:

> *It allows me to take the music with me whereever I go. The same music as I was listening to [in the previous room].* (A1)

This illustrates that both families found the feature useful and both families had a good understanding of how the fuctioned worked.

When the participants were asked about their thoughts on the functionality of Movement, they were excited. In fact, they wished their own music setups contained similar features:

> *I definitely think it was useful. It is almost going to be sad to miss outtely on it now that I have had access to the function for some weeks. I definitely think it has potential.* (B1)

The enthusiasm was shared amongst the participants from household A:

> *It was really clever. It was close to invaluable.* (A1)

> *Yes it was really smart.* (A2)

Besides showing their enthusiasm towards the Movement function, it also shows that even though household A at times experienced unintended behavior, they still thought it was a brilliant idea, and they would have liked to see similar functionality in their Sonos system.

5.2.2 Quantitative Data Analysis

Besides the qualitative data obtained through the interviews with the test participants, a large amount of quantitative data was collected through the logging functionality of AirPlayer. This section contains an analysis of the collected log data from both of the participating families, and the results are presented and compared to the findings obtained through the analysis of the qualitative data. The sample data of the log files from the two families exceeds 78 gigabytes of logged data divided between the two log files.

By examining the log files, general usage patterns of the system from both families were extracted. Figure 5.2 and Figure 5.3 provides a visual representation of the usage of AirPlayer during the 21 days of testing for household A and household B respectively.

As the charts show, the participants from household A used the system mostly during the first half of the testing period, averaging 3.3 hours per day throughout the test period. Household B, on the other hand, had a more even distribution of usage throughout the test period, but only for an average duration of 2.0 hours per day.

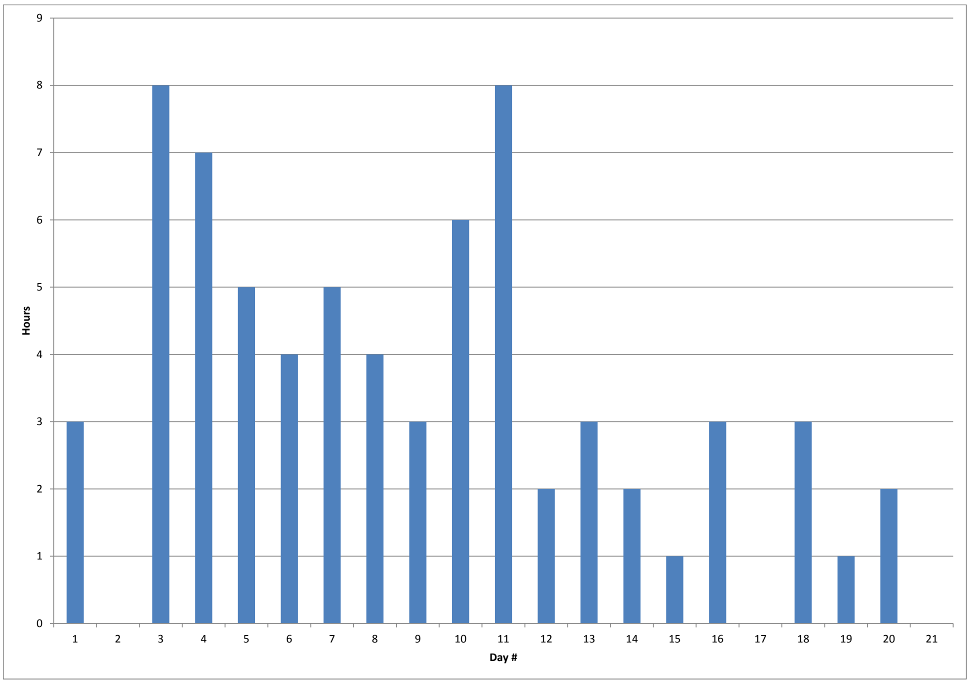


Figure 5.2: General usage of AirPlayer in hours per day for household A

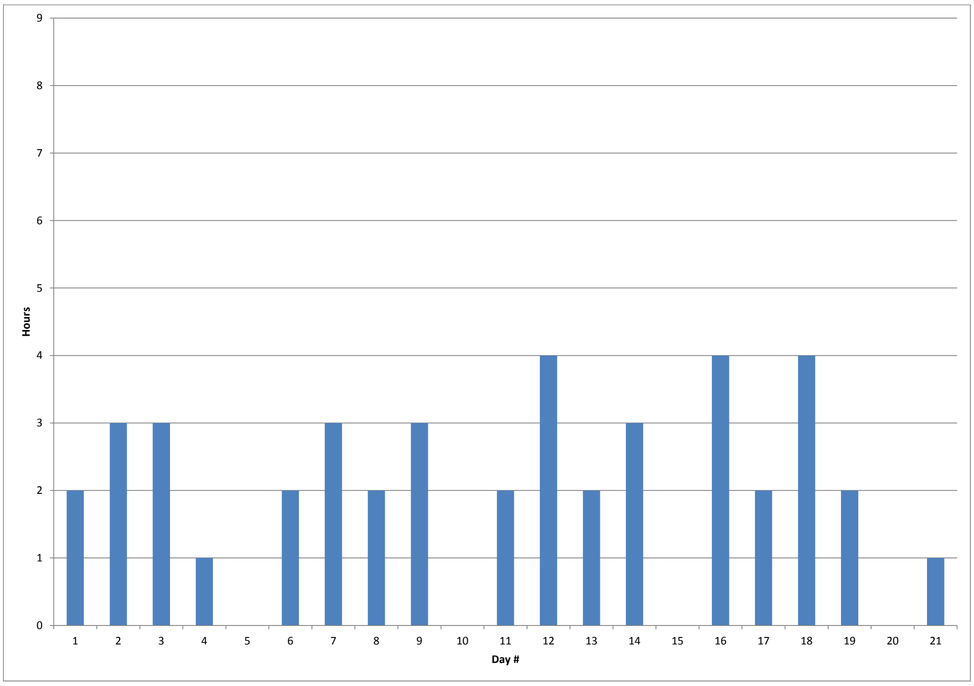


Figure 5.3: General usage of AirPlayer in hours per day for household B

5.2.2.1 Location Functionality

The usage of the Location functionality of AirPlayer was a key area of focus for the field study, and usage patterns of the Location functionality for both families were extracted from the logged data and analyzed. Figure 5.4 and Figure 5.5 shows the percentage of the total time per day in which the Location function was enabled in AirPlayer for each household. The percentage values were compiled by comparing how long a function was enabled compared to the total system uptime. The total system uptime is defined by the time between the first user interaction and until system termination, that is, until the system was shut down by the user. While both Location and Movement can be enabled concurrently, the Location functionality has no effect in the system while the Movement functionality is active. Thus, the intervals in which both functionalities were active have not been included in the calculations for the Location functionality.

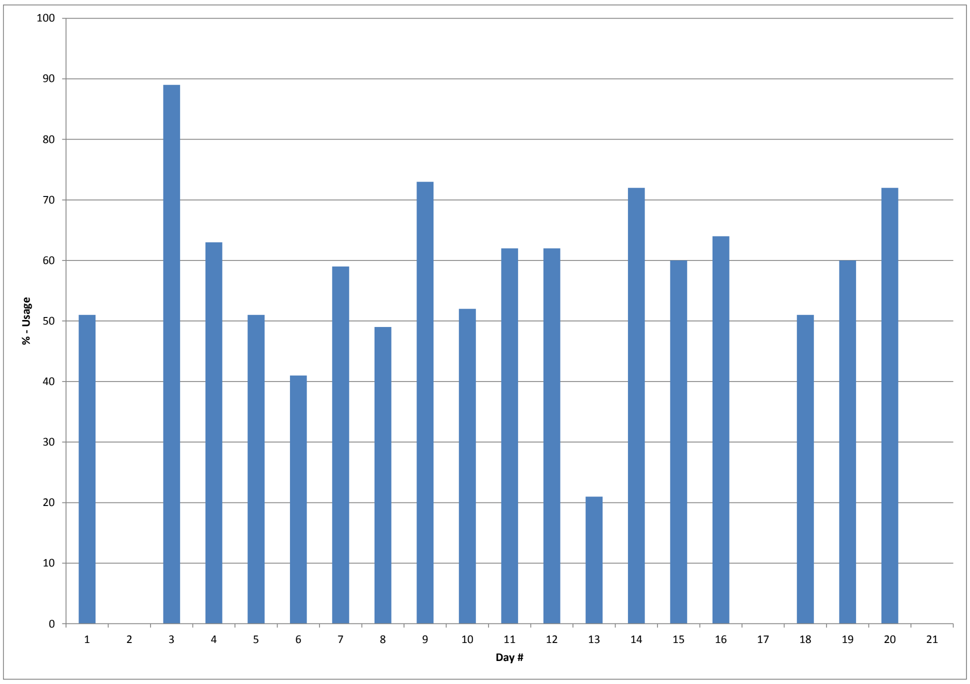


Figure 5.4: Usage of the Location function for household A based upon time used in relation to the hours used on the individual days.

As shown in the charts of Figure 5.4 and Figure 5.5, the Location function was enabled for a significant part of the testing period. Specifically, the function was enabled for 57 percent and 47 percent in average of the time in household A and household B respectively. This observation supports the claims by both families in that they both expressed extensive use of this function, as they found the function to be both useful, intuitive and effective. The logged data from household A shows an almost even distribution of time used throughout the period, whereas household B used the Location function a lot at the beginning of the period, and barely used it at the end of the testing period.

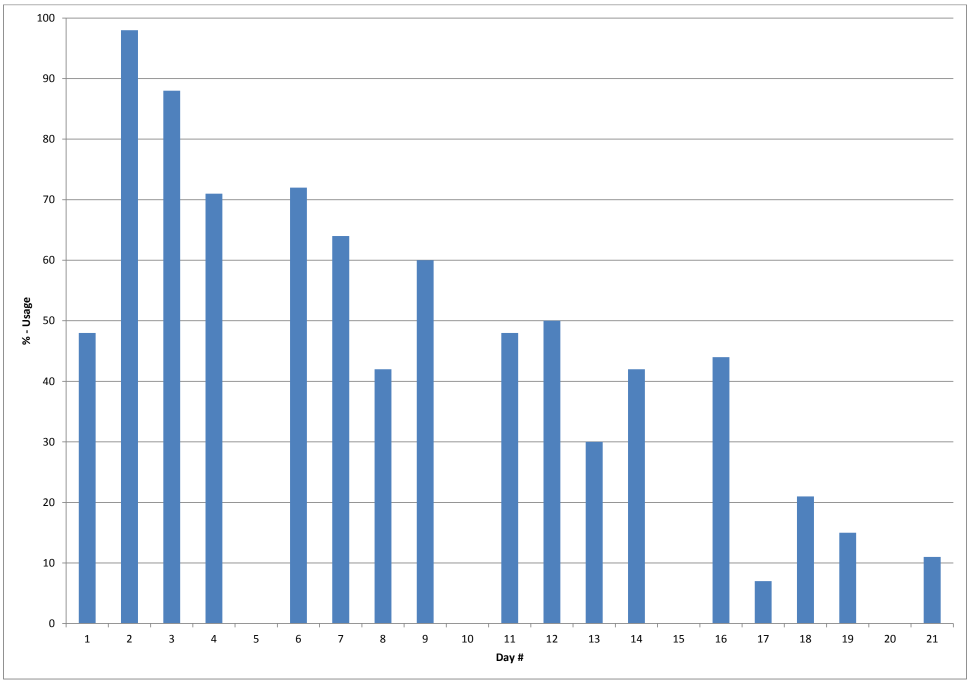


Figure 5.5: Usage of the Location function for household B based upon time used in relation to the hours used on the individual days.

5.2.2.2 Movement Functionality

Another key area of focus for the field study was the usage of the Movement functionality, and the usage patterns of the Movement functionality were extracted from the logged data and analyzed for both households. Figure 5.6 and Figure 5.7 show the percentage of which the Movement function was enabled throughout test period for both participating households.

During the interviews, the participants expressed that they were using the Movement function most of the time in the interaction with AirPlayer, however, by examining the log files it was found that the functionality was used for only 23 and 39 percent of the time at household A and B respectively. Figure 5.6 and Figure 5.7, shows that the behavior of the participants from household A resembles the behavior observed in the usage of the Location functionality, with an even distribution of usage of the Movement functionality throughout the test period Similarly, the usage pattern of household B reveals a relation to the usage of the Location functionality, which indicates that the test participants of household B shifted focus from testing the Location functionality in the beginning of the test period, and put more focus on the Movement functionality at the end of the testing period.

An interesting observation shows a correlation between the the usage of the Movement functionality and the amount of interactions with the system over time. Figure 5.8 and Figure 5.9 show the average amount of interactions per hour throughout the test period. The figures illustrate that both households used a varied amount of interactions on average. However, when comparing these charts to the charts on Figure 5.6 and Figure 5.7, it indicates a gen-

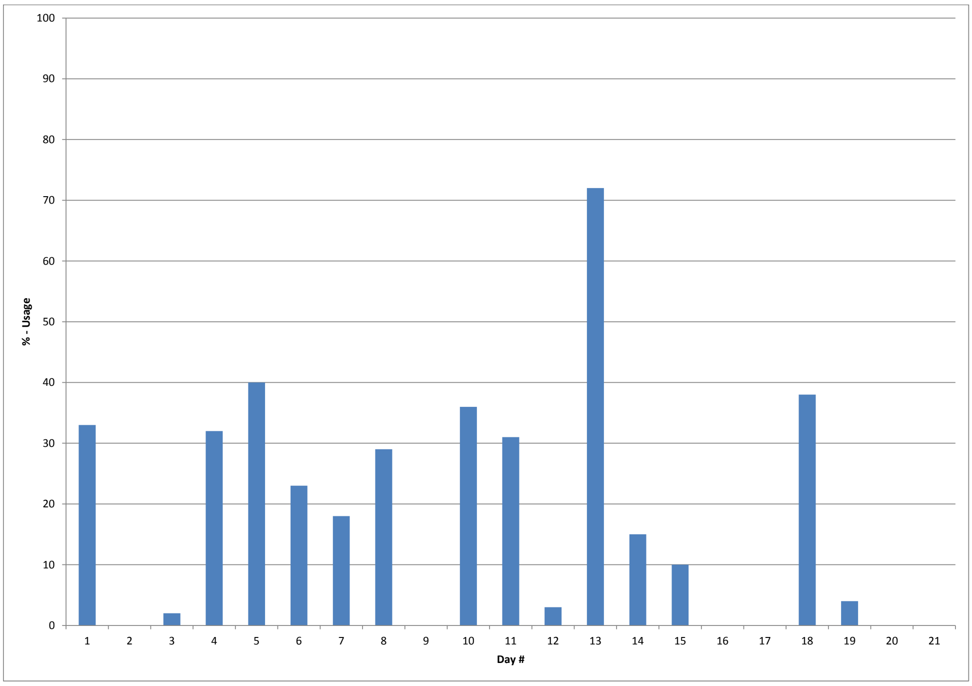


Figure 5.6: Usage of the Movement function for household A based upon time used in relation to the hours used on the individual days

eral pattern that the amount of interactions have a tendency to decrease when the Movement functionality is enabled. By reviewing the logged data a tendency was revealed as a reduction of interactions per hour of 42 percent on average when the Movement functionality is enabled.

5.2.2.3 Usage of the Panic Button

A noteworthy side observation which was revealed through both the interviews and the logged data, shows that the panic button proved to be more useful in the system than anticipated. Both families mentioned the functionality as a highly useful feature to restore the system to its original state. Household A even expressed a desire for having such function in their own Sonos multi-room music system.

Figure 5.10 and Figure 5.11 shows the usage of the panic functionality over time for household A and B respectively.

As Figure 5.10 shows, household A has made little use of the panic button throughout the initial phase of the testing period. An inspection of the log file entries reveal that the system did not encounter any errors during this period. However, after a period of time, the system encountered an error, which according to the log files were accounted for by using the panic functionality. This restored the system back to its original state, and allowed the user to continue using the system. From this point, log file entries reveal that additional usage of the panic button were primarily unrelated to errors in the application, but as the participants expressed, they simply used it to restart the system.

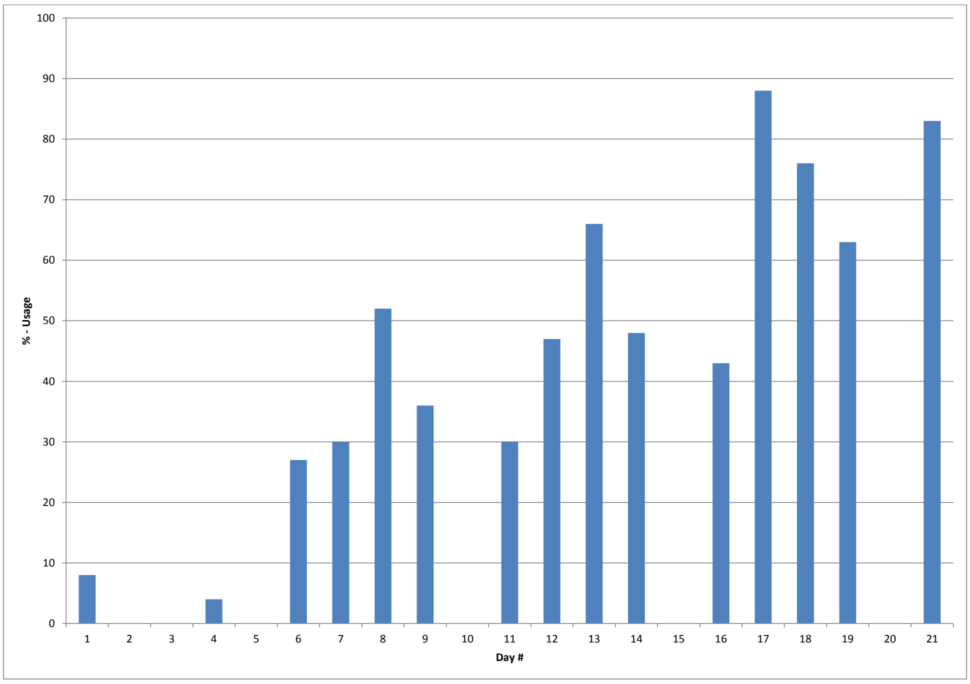


Figure 5.7: Usage of the Movement function for household B based upon time used in relation to the hours used on the individual days

Similar behavior can be observed from household B, as the usage of the panic button was not related to errors in the Android application. Figure 5.11 shows the data obtained from the log file from household B.

Both figures show that while no families used the panic button at first, it would seem that after they discovered the exact result of using the button, they would continue to use the panic button to reset the system as opposed to its original purpose of accounting for system errors.

5.3 Summary

AirPlayer was deployed for testing in an in-situ field study to evaluate the integration of the proxemic dimensions in the interaction design. The field study was conducted at home by the participating test families, where the system was installed and integrated with the participants' existing multi-room music systems over a period of three weeks. Also, the test period was concluded with an interview where the test participants were asked to elaborate on their experience with the system.

Through the field study it was revealed that the test participants generally appreciated the integration of the proxemic dimensions in the interaction design of AirPlayer. All test participants had a good understanding of how the functionalities were supposed to work, and were satisfied by the assistance that AirPlayer provided in relation to interacting with the system.

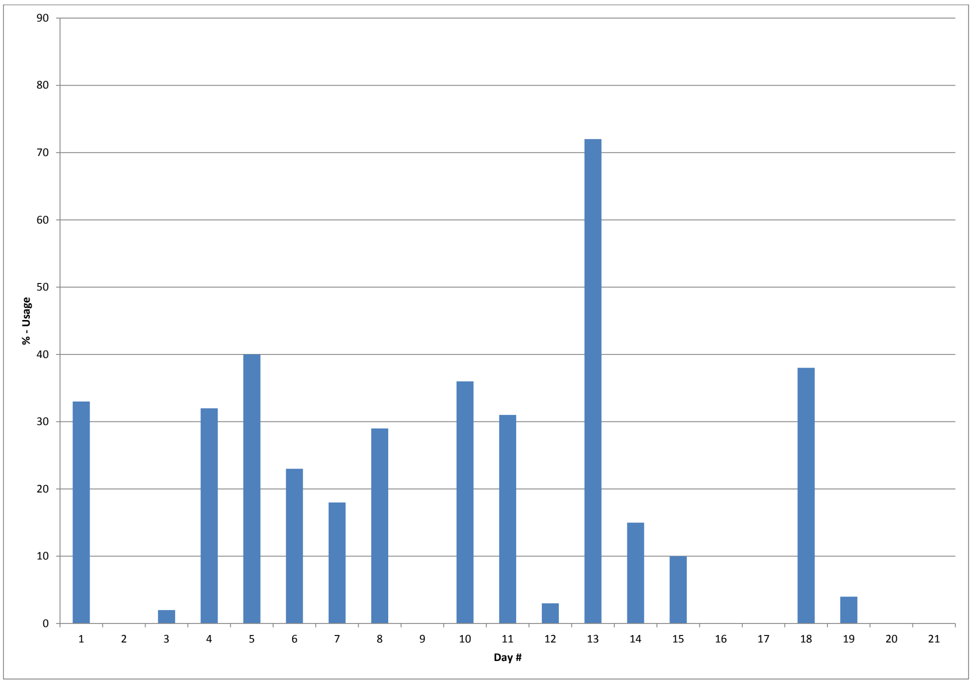


Figure 5.8: Average amount of interactions per hour throughout the test period for household A

The data collected through the automatic logging function of AirPlayer supports these statements from the users, and reveals the functions were rapidly utilized throughout the testing period, i.e. the Location functionality was active for 57% of the time in household A and 47% of the time in household B, while the Movement functionality was active for 23% and 39% of the time.

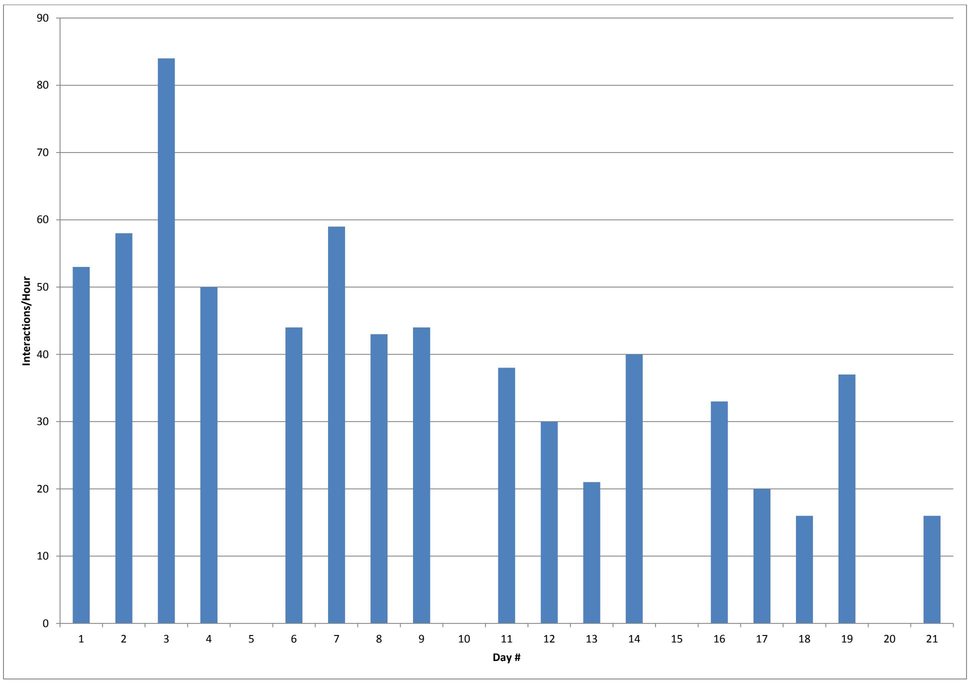


Figure 5.9: Average amount of interactions per hour throughout the test period for household B

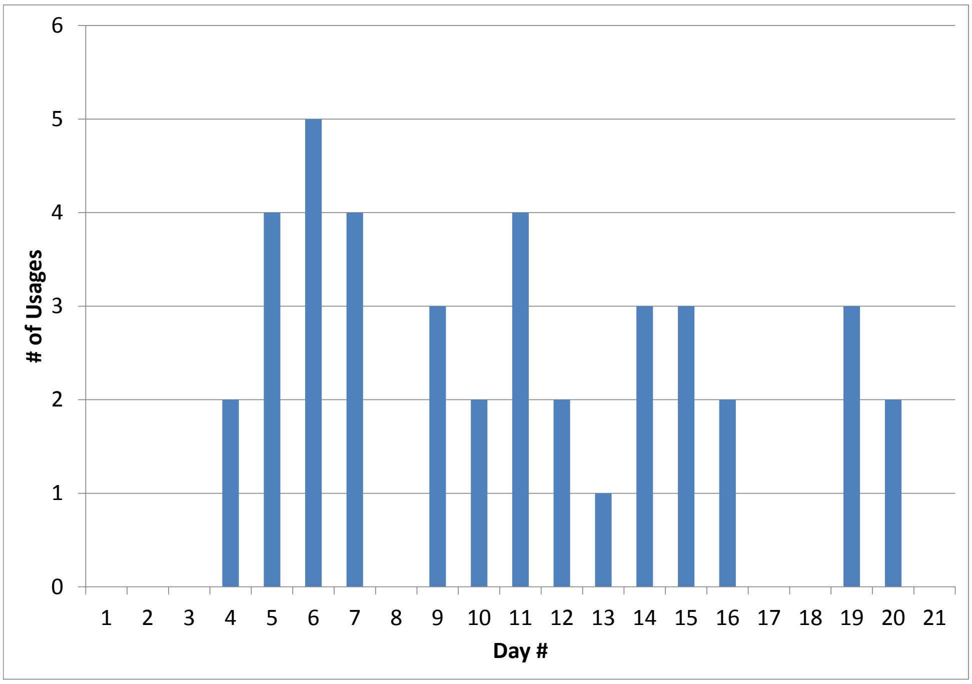


Figure 5.10: Usage of the panic button for household A throughout the 21 day test period

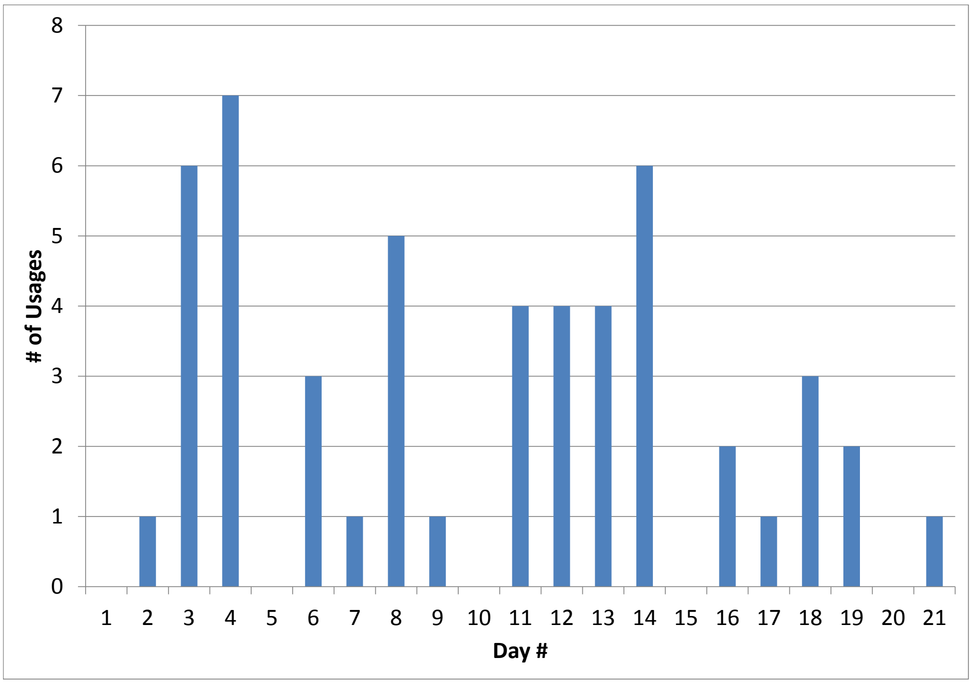


Figure 5.11: Usage of the panic button for household B throughout the 21 day test period

CHAPTER 6

DISCUSSION

Based on previous studies in the area of proxemic interaction, AirPlayer was developed as an attempt to integrate the proxemic principles into the domain of multi-room music systems.

The development of AirPlayer as a prototype multi-room music system has resulted in a set of interesting findings as described in Chapter 5. This chapter contains the discussion based on the findings from the test period.

6.1 Proxemic Interaction

The focus for this master thesis was to explore the possibilities with proxemic interaction principles in a multi-room music system. These possibilities were explored through the development of AirPlayer, which was then installed for testing in a field study within two households. The remainder of this section covers the two implemented proxemic dimensions, that is, they discuss how they were used and how the users perceived the functions, in order to estimate whether or not Location and Movement were found useful in AirPlayer.

The findings from both the interviews and the log files revealed that the Location functionality had been well received by both households. During the interviews, the participants expressed that they found the function to be intuitive, useful and transparent. The function itself was used for the majority of the time, that is, the function was used for 57% and 47% of the total time by household A and household B respectively. According to our data, we did not find a connection between the amounts of interactions and having enabled Location. The reason for this is most like due to the design of the feature, as it enables the application to automatically control the closest zone, and therefore effectively removes a few interactions in an environment like the participating households. However in relation to the total amount of interactions, this did not significantly impact the overall result. Furthermore, during the test period neither of the participating households encountered any errors when using the Location function nor did the log files reveal any troubles using it from a system perspective. In addition to this, both of the participating families expressed to have a good understanding of how the function worked.

The most anticipated feature amongst the test participants was the Movement functionality, as it is differs from features known from their own multi-room music systems. During the interviews, both families expressed that they had used Movement for a large portion of the time, however examining the log files reveals that they had used it for 23% and 39% at household A and B respectively. Closer inspection of the data from the log files reveals a pattern for both households, indicating that in the period where participants were working from home or were otherwise using the system during work hours, and thereby presumable were alone with the system, the Movement function was enabled almost the entire time. The conducted interviews supports this observation, as both household A and household B describe how they used the system in working situations from home. By design, the Movement functionality mainly plays in a subset of the zones, thereby creating an environment where the music is limited to few rooms, and therefore fits situations where people are located within the same zones.

In contrast to Location, the findings from the log files indicated that the amount of direct interactions are reduced when Movement had been enabled. In fact, when Movement was enabled, we found that on average, 42% less interactions were performed in comparison to when it was not enabled. An observation in the log files reveals that while Movement was enabled, the participants did not have the same amount of interactions when walking in between the different zones as a result of not having to set up each zone as they were entered. Furthermore, the participants did not need to stop or reset particular zones as the function would do this automatically.

As a result of the narrow data foundation built through the field study, it is hard to come to a general conclusion regarding the usefulness of the two proxemic dimensions in the interaction design of a multi-room music system. However both the participating households expressed that they found Location and Movement to be useful and satisfactory to use, and that they would have liked to have the possibility of integrating the functionalities in their existing multi-room music systems.

6.2 Usage Over Time

As a part of our in-situ test, we expected to see a decline of interest over time from the participating households as a result of having the participants getting familiar with the system, and that they would therefore consider the system to be less exciting and challenging [8]. In fact, the original plan for the testing period was to divide the three weeks of testing into three different phases, each with a different focus; the first week would include getting familiar with the system, the next would focus on the Location functionality and the last would focus on Movement. However, we did not want to risk of having the participants to lose interest in the system, which is reflected by the chosen solution for the test period. The data gathered through the testing period shows that while household B used the AirPlayer system in a similar manner throughout the period, the usage of the system did indeed decline over time as predicted. However, it is not known whether this is caused by them finding the system to becoming trivial at the end, or if other outside parametres influenced their use of the system.

In relation to getting feedback on the proxemic dimensions, both participating households chose to test them differently according to the log files. As our findings showed, household A chose to use both functions the same way during the test period, whereas the gathered data indicates that household B changed focus from using Location at the begining to using Movement at the end. While the testing period did not contain strict instructions on which features to test when, both of these approaches from the participants provided valuable insights in the usage of the dimensions despite of their individual ways of doing things.

6.3 Narrow Data Foundation

As previously mentioned, the collected data set was not sufficient to make general conclusions from the usage of AirPlayer. The field study was conducted at home by two participating test families, where AirPlayer was installed through a period of three weeks. Throughout the testing period, all events and interaction points in the system was automatically logged in a log file, and the test period was concluded with an interview with the participating test families. While the field study resulted in a large amount of data, the data is based on only two data sources, i.e. the two participating families. Furthermore, the two participating families showed very different usage patterns of the system, and thus, only a few similar tendencies were observed. This means that the collected data is inconclusive in relation to general usefulness of the explored functionalities, and should only be used as indications to reveal potential areas of interest for later studies within the area.

Ahead of the test period, we planned to test AirPlayer on a group of three to five families, however it was difficult to find families who wished to participate, and spend their spare time testing the system. While this is still a narrow audience, it could have provided an even better overview of the general usefulness of the tested functionalities.

CHAPTER 7

CONCLUSION

This master thesis has evolved around establishing how the proxemic principles can be applied to the interaction design of a multi-room music system. The primary research question was defined as follows:

> *How can the principles of proxemic interaction assist users in the interaction with multi-room music systems?*

This research question has been answered through the development of a prototype multi-room music system called AirPlayer. AirPlayer is a music system which supports playback of music in individual zones in an environment. Each zone can be controlled individually, or zones can be linked to be controlled as a single unity. The system has support for proxemic interaction principles, that is, the proxemity of entities in the environment defines the interaction with the system.

The master thesis is based on results obtained and documented in the previous semesters project report.

During the previous semester project, the concept of AirPlayer was defined and the basic requirements were specified. Through a market research and a technological analysis of existing multi-room music systems, the basic requirements for AirPlayer were defined. Furthermore, through a study of related work within the fields of proxemic interaction and ubiquitous music, the possibilities to integrate the proxemic dimensions into a multi-room music system environment were explored. It was found that none of the studied existing systems were able to support proxemic interaction, and thus a technical framework was developed from scratch on top of the AirPlay protocol [2].

The technical framework formed the foundation for the implementation of a prototype with support for proxemic interaction principles during the period of this master thesis. The remote control was implemented as a smartphone application, allowing users to control their music system from whereever in the environment they are located. Using a simple implementation of Wifi triangulation, the application approximated the users position within the environment and the results were used in the implementation of two proxemic dimensions, i.e. Location and Movement. The proxemic dimensions were assisting the users when interacting with AirPlayer, that is, the Location dimensions would track the users current location, and present a remote interface accordingly and the Movement dimensions allowed the user to have the music following him around the environment, thereby ensuring a continous music experience.

After the implementation, AirPlayer was deployed for testing in a real user environment. A set of selected test participants had the system installed and used it in their everyday life in a period spanning over three weeks. During the test period, the test participants recorded their thourghts about the system in a log. Furthermore, the usage of AirPlayer was automatically logged by the system. Lastly, at the end of the test period, the participants were interviewed regarding their experiences with the system. The collected data were analysed and created the basis for answering the primary research question.

Through the field study it was revealed that the users generally appreciated the integration of the proxemic dimensions in the interaction design of AirPlayer. The findings were based on a narrow data foundation, thus no general tendencies can be concluded from the collected data, however, the observations indicates that the proxemic dimension assisted the users when completing general tasks of managing the multi-room music system, i.e. control the music playback of a particular zone or change the music between zones. While the field study only revealed indications of general tendencies between the users, the results can form the foundation for future studies within the area of proxemic interaction.

BIBLIOGRAPHY

[1] Apple Inc. Apple Store. http://store.apple.com/, October 2011.

[2] Apple Inc. AirPlay. http://www.apple.com/dk/itunes/airplay/, May 2012.

[3] Apple Inc. iTunes. http://www.apple.com/itunes/, May 2012.

[4] Till Ballendat, Nicolai Marquardt, and Saul Greenberg. Proxemic Interaction: Designing for a Priximity and Orientation-Aware Environment. *ITS '10: ACM International Conference on Interactive Tabletops and Surfaces*, pages 121–130, November 2010.

[5] Bang & Olufsen. BeoLink - move the experience from room to room. http://www.bang-olufsen.com/home-integration-beolink, October 2011.

[6] Bose. Bose Link. http://worldwide.bose.com/axa/en_au/web/bose_link_technology/page.html, October 2011.

[7] Adrian Clark, Andreas DÃ¼nser, Mark Billinghurst, Thammathip Piumsomboon, and David Altimira. Seamless Interaction in Space. *Proceedings of the 23rd Australian Computer-Human Interaction Conference*, pages 88,97, 2011.

[8] Katiee Crowley, Aidan Sliney, Ian Pitt, and Dave Murphy. Capturing and using emotion-based BCI signals in experiments; how subject's effort can influence results. *Proceedings of HCI 2011*, pages 132–138, 2011.

[9] Rob Diaz-Marino and Saul Greenberg. The Proximity Toolkit and ViconFace: The Video. *CHI2010*, pages 4793–4797, April 2010.

[10] Kasper Gadensgaard and Mathies Grøndahl Kristensen. AirPlayer - Proxemic Interaction in Multi-Room Music Systems, December 2011.

[11] Saul Greenberg. Opportunities for Proxemic Interactions in Ubicomp. *INTERACT 2011 Part I*, pages 3–10, 2011.

[12] Saul Greenberg, Nicolai Marquardy, Till Ballendat, Rob Diaz-Marino, and Miaosen Wang. Proxemic Interactions: The New Ubicomp? *Magazine: interactions - Proxemic Interactions: The New Ubicomp?*, XVIII(1):42–50, January 2011.

[13] Edward Twitchell Hall. *The Hidden Dimension*. Anchor Books, reprint edition, 1990.

[14] Lars Erik Holmquist. Ubiquitous music. *Magazine: interactions - Ambient intelligence: exploring our living environment*, 12(4):71–78, 2005.

[15] Jon Lech Johansen. JustePort. http://nanocr.eu/software/justeport/, July 2005.

[16] Jonathan Lazar, Jinjuan Heidi Fend, and Harry Hochheiser. *Research Methods In Human-Computer Interaction.* Wiley, 1st edition, 2010.

[17] Lassi Liikkanen and Mikko Lahdensuo. Observing the Mobile Music Phenomenon: One in Nine Commuters is Wired. *DIS '10: Proceedings of the 8th ACM Conference on Designing Interactive Systems*, August 2010.

[18] Nicolai Marquardt. Proxemic Interactions in Ubiquitous Computing Ecologies. *CHI2011*, pages 1033–1036, May 2011.

[19] Nicolai Marquardt, Robert Diaz-Marino, Sebastian Boring, and Saul Greenberg. The Proximity Toolkit: Prototyping Proxemic Interactions in Ubiquitous Computing Ecologies. *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 315–325, 2011.

[20] Marcelo Soares Pimenta, Luciano Vargas Flores, Ariadna Capasso, Patricia Tinajero, and DamiÃ¡n Keller. Ubiquitous Music: Concepts and Metaphors. *Proceedings of the XII Brazilian Symposium on Computer Music*, 2009.

[21] Sonos Inc. Sonos Wireless Hifi System. http://www.sonos.com/system/, October 2011.

[22] Mark Weiser. Ubiquitous Computing. http://www.ubiq.com/hypertext/weiser/UbiCompHotTopics.html, August 1993.

APPENDIX A

INTERVIEW GUIDE

Formalia

- I har i de seneste uger haft vores program AirPlayer installeret i jeres hjem, og i den forbindelse vil vi gerne interviewe jer omkring jeres oplevelse med programmet.
- Formålet med interviewet er at afdække hvorvidt vores implementerede interaktionsforslag hjælper brugerne (Jer) i forbindelse med interaktionen mellem brugere og multirumsmusiksystemer.
- I den forbindelse leder vi ikke efter et specifikt resultat, men en afklaring af fordele og ulemper ved denne nye interaktionsform. Der er derfor ikke nogle svar der er mere rigtige end andre.
- Vi vil gerne optage interviewet. Optagelserne bliver kun brugt internt til transkribering. Transkriberingen af interviewet vil indgå som en del af studierapporten.

Generelt

- Først vil vi gerne høre hvad i overordnet synes om systemet.
  - Fungerede det som forventet?
  - Var der væsentlige mangler i forhold til jeres nuværende system, osv.?
  - Var der noget i kunne have ønsket anderledes?
- Hvor meget har i ca anvendt systemet?
  - I hvilke situationer har i anvendt systemet?
  - Hvordan har i anvendt systemet i den givne situation?

- Oplevede i problemer i forbindelse med systemet i perioden?
  - Hvad fungerede ikke?
  - Var det af afgørende betydning for oplevelsen?
- Efter Jeres opfattelse, hvordan adskilte vores system sig fra det eksisterende system?
  - Er der nogle bestemte funktioner i systemet som adskiller sig og hvordan fungerede de?

Location

Vi har implementeret denne funktion, Location, som aktiveres via knappen øverst til venstre.

- Beskriv med Jeres ord hvad funktionen gør i systemet.
- Har I anvendt funktionen?
  - Hvornår?
  - Hvor meget?
  - Hvorfor så meget/lidt?
- Var I i tvivl om selve anvendelse af funktionen?
  - Var det forståeligt hvad funktionen gjorde og hvorfor den gjorde det?
  - Var der perioder hvor systemet ikke opførte sig som forventet?
    * Hvad skete der?
    * Hvad forventede I?
- Hvad synes I om funktionen?
  - Er den brugbar eller overflødig?

Movement

Vi har implementeret denne funktion, Movement, som aktiveres via knappen øverst til højre.

- Beskriv med Jeres ord hvad funktionen gør i systemet.
- Har I anvendt funktionen?
  - Hvornår?
  - Hvor meget?
  - Hvorfor så meget/lidt?
- Var I i tvivl om selve anvendelse af funktionen?

  - Var det forståeligt hvad funktionen gjorde og hvorfor den gjorde det?
  - Var der perioder hvor systemet ikke opførte sig som forventet?
    * Hvad skete der?
    * Hvad forventede I?
- Hvad synes I om funktionen?
  - Er den brugbar eller overflødig?

APPENDIX B

Interview with Family A

Interviewer: Først vil jeg gerne høre Jer hvad i overordnet synes om systemet med Jeres egne ord? Fungerer det som forventet?

A1: Tanken bagved systemet er rigtig godt, men jeg mangler et par små finesser. Vi har sat Kim Larsen til at køre, men efter to døgn opdager vi at vi har hørt det til hudløshed fordi den kører det hele om igen. Jeg troede den stoppede igen når der ikke var flere numre.

Interviewer: Så det er fordi den automatisk kører på repeat?

A1: Ja.

Interviewer: Der kunne i godt have ønsket en funktion man kunne så til og fra?

A1: Ja. Og hvis man valgte noget musik man gerne ville have skulle komme efter [det musik der var i køen], så fortalte den bare at spillelisten var fuld, og det var selvfølgelig fordi at ingen af sangene blev fjernet fra køen.

Interviewer: Så når et nummer var færdigafspillet, så ville I gerne have fjernet den fra musiklisten?

A1: Ja.

Interviewer: Var der nogle væsentlige mangler i dette system i forhold til Jeres Sonos system?

A1: Der manglede vi jo radioen. Rigtig meget. Men til gengæld synes jeg det var en fed idé at musikken følger én.

A1: Jeg havde troet når jeg gik ovenpå, og gik i seng, at musikken stoppede. Det gjorde den ikke. Den blev i det sidste rum jeg forlod - det undrede mig.

Interviewer: Okay, så du ville gerne have at når man kom uden for rækkevidde at musikken selv stoppede helt?

A1: Ja, så skulle den slukke. Det ville være nemt at man vidste det hele var slukket når man gik i seng, og lige sådan ville det være dejligt at når man så stod op om morgenen, ja så startede det igen.

A2: Rækkevidden dækker vel også fra soveværelset og herned til alrummet, gør den ikke det?

A1: Jo det kan godt være at det er fordi at de er for tætte på rummet. Det kan godt være.

Interviewer: Var der noget i kunne ønske var anderledes i forhold til systemet ud over de ting I allerede har nævnt? Var der noget der overraskede Jer når i brugte det?

A1: Jeg tror at selve senderne de måske sad for tætte. Den var lidt lang tid om at finde hvilken sender den skulle finde når man gik fra rum til rum. Den gik egentlig helt ud inden den kom igang igen.

Interviewer: Okay.

A2: Når den først havde fundet dem, så gik det hurtigt, synes jeg. Altså når den først havde fundet hvilket rum man var I, så kunne man nemt gå fra rum til rum.

Interviewer: Var det når musikken fulgte efter Jer? eller var det i forbindelse med automatisk valg af nærmeste zone?

A2: Når musikken fulgte efter mig.

Interviewer: Okay. Men det var ikke den opfattelse du havde [Interviewer spørger A1]

A1: Nej jeg synes den var lang tid om at skifte. Nogle dage. Ikke alle dage. Andre dage virkede det godt.

Interviewer: Ja okay. Hvor meget har i brugt systemet imens det har været her? Hvor mange timer om ugen eller om dagen?

A1: Vi har været nødt til at genstarte systemet et par gange undervejs, men ellers har vi nogle dage haft det kørende hele dage, og andre dage fra et til to stykker om aftenen til sengetid. Så det er meget forskelligt.

Interviewer: Er det nogle timer om ugen eller nogle timer om dagen?

A1: Det har minimum været et par timer om dagen.

Interviewer: Okay.

Interviewer: I hvilke situationer har I anvendt systemet? Har det været når I har haft gæster, når I selv har gået derhjemme eller har det primært været for at teste systemet?

A1: Det har været for at teste systemet, men det har mest bare været når vi var derhjemme os selv, så vi kunne teste det. Altså teste hvor langt vi kunne gå væk før det holdt op med at spille når vi tog fra rum til rum. Vi har ikke brugt det når vi har haft gæster.

Interviewer: Har i anvendt det som en del af Jeres dagligdag som Jeres gamle system? Eller har I anvendt det fordi vi gerne ville have at I skulle teste systemet?

A1: Det har været ligesom det gamle systemet, og så engang imellem fordi vi skulle teste. Det med at flytte fra rum til rum det duer ikke når vi bare går derhjemme, for går jeg og gør rent, så hopper den jo hele tiden. Men hvis man er i et rum og går ned på kontoret og fortsætter på kontoret så er det jo fint. De dage hvor vi bare har brugt det hjemme, der har vi bare brugt det og gået fra rum til rum, mest bare almindeligt brug.

Interviewer: Nu siger du så at når du går og gør rent, hvordan har I så anvendt systemet? Altså når du gør rent så siger du at du ikke bruger funktionen hvor musikken følger dig.

A1: Ja så har jeg bare tændt musikken i hele huset fordi så har jeg musik ligemeget hvor jeg er. Og ellers, så hvis vi bare har været i køkkenet, jamen så har vi bare haft musikken tændt i køkkenet, og lige sådan på kontoret.

Interviewer: Okay. Du har tidligere nævnt at I har oplevet nogle problemer med at systemet ikke fungerede. Hvad fungerede ikke? Opdagede i det?

A1: Det ved du. Det var dig der oplevede det. [Ser på A2]

A2: Det var bare den popup der. Den poppede op og så gik jeg ind og lukkede den, og så genstartede jeg Appen og så virkede det.

Interviewer: Okay, så det var en popup i iTunes der gav problemer?

A2: Ja.

Interviewer: Så det var ikke en fejl i forbindelse med systemet?

A2: Nej.

Interviewer: I har ikke oplevet problemer med at finde rummene i systemet?

A2: Hvis vi har haft nogle problemer, så har vi bare brugt panikknappen, og så virkede det.

Interviewer: Okay, så I har brugt panikknappen?

A2: Ja.

A1: Ja det var en fed funktion.

Interviewer: De problemer I har oplevet, var de af afgørende betydning for den måde i oplevede systemet?

A1: Panikknappen gjorde at vi selv har kunne fikse problemerne. Havde vi ikke have haft den, så ville vi have haft problemer.

Interviewer: Så der har ikke været noget der har ødelagt musikoplevelsen i forbindelse med systemet?

A1: Ork nej.

Interviewer: Okay.

Interviewer: Efter Jeres opfattelse, hvordan adskiller vores system sig fra det system i havde i forvejen.

A2: Altså jeg er ikke så god til at bruge Android.

Interviewer: Så det var simpelthen et spørgsmål om tilvænning?

A2: ja hvis man havde en Android, så var det nemt nok.

Interviewer: Men ellers musiksystemet i forhold til Sonos?

A2: Det er sådan set det samme, og så har I den ekstrafunktion hvor man kan se hvor man er og få musikken til at følge efter.

A1: Jeg kunne godt tænke mig at vi havde panikknappen i Sonos. Den kunne jeg godt bruge den anden vej, men det var ikke det du spurgte om. Det var rart at have panikknappen hvor man kunne starte forfra.

Interviewer: Så den kunne i godt bruge i Jeres Sonos-system?

A1: Ja.

Interviewer: Hvordan opfattede i de andre funktioner?

A1: Jeg synes det er rart at den slukkede og tændte i rummene når vi brugte det. Altså at man kunne få den til at følge én.

Interviewer: Ja.

Interviewer: Vi har implementeret to funktioner som vores studie handler om. Den ene hedder location og den anden hedder movement. I location, som er den funktion som aktiveres øverst til venstre [Interviewer peger på ikonet] på den lille pil, så fjernbetjeningen skifter i forhold til det rum man befinder sig i.

A1: Ja.

Interviewer: Vil i sætte ord på hvordan I opfattede den fungerede?

A2: Det fungerer som det skal. Den gør at vi nemt kan styre musikken og finde det rum man befinder sig i. Så behøver man ikke selv finde rummet.

Interviewer: Er det en funktion I har benyttet?

A2: Ikke lige så meget som den hvor musikken følger én.

A1: Altså for mig må den have virket fantastisk, fordi jeg brugte den og jeg tænkte ikke på at jeg havde koblet den til. Jeg troede egentlig at den var koblet sammen med den anden hvor musikken fulgte mig fra rum til rum. Den virkede jo når jeg skruede op og ned og hvad jeg ellers fandt på. Men jeg har ikke tænkt på jeg har brugt den, men det har jeg jo.

Interviewer: Hvor meget har du brugt den? er det hver gang du har brugt systemet?

A1: Jeg har altid brugt den, jeg har altid haft den slået til. Altså når jeg har været et par timer i køkkenet og er gået ind på kontoret. Der har jeg jo [fjernbetjeningen] med. Og der har jeg kunne styre musikken der hvor jeg var.

Interviewer: Okay.

Interviewer: Var I i tvivl om hvordan man anvendte funktionen?

A1: Det varede lidt inden jeg forstod dens funktion og hvad jeg skulle bruge den til. Jeg kunne godt have ønsket en lidt bedre introduktion når der var så nyt. Det varede lidt før det gik op for en at det var smart.

Interviewer: Var der perioder hvor denne funktion ikke opførte sig som den skulle?

A2: Ikke hvad jeg har oplevet.

A1: Nej det tror jeg ikke. Nej den har opført sig som den skulle.

Interviewer: Okay.

Interviewer: Den anden funktion er movement, som er den funktion hvor musikken følger Jer rundt, altså Jeres bevægelser, som kunne aktiveres fra den her knap øverst til højre. [Interviewer peger på knappen]. Vil i beskrive med Jeres ord hvad den gør i systemet?

A1: Den gør at jeg tager musikken med rundt der hvor jeg er, det samme musik ligemeget hvor jeg er. Den var også rigtig brugbar, men der skulle måske have været lidt mere overlapningstid.

Interviewer: Okay. Så den tænder for sent og slukker for tidligt?

A1: Den kunne godt have ventet 15 sekunder med at slukke i det rum man forlader, og så tænde med det samme der hvor man går hen i det nye rum. Så lidt overlapning ville have gjort underværker.

A2: Ja det ville have været smart.

Interviewer: Er det en funktion I har benyttet?

A1 + A2: Ja.

Interviewer: Hvor meget har i benyttet den?

A1: Tit. Hver gang faktisk.

A2: Jeg har mest bare leget med [funktionen] for at teste den.

Interviewer: Hvornår har i brugt den? I hvilken forbindelse?

A1: Jeg har brugt den når jeg har været de forskellige steder. Altså jeg har brugt den når jeg er gået fra privaten til kontoret, men jeg slog den fra når jeg gjorde rent, fordi der var det uhensigtsmæssigt.

Interviewer: Og det var fordi?

A1: Jamen det var fordi den ikke havde de der 15 sekunder, så blev det lidt irriterende.

Interviewer: Okay.

Interviewer: Var I i tvivl om hvordan denne funktion fungerede? Eller hvad meningen var med den?

A1: Nej. Altså meningen var jo at musikken skulle spille hvor jeg var. Det må have været meningen med funktionen.

Interviewer: Og havde det været klart fra start at det var formålet med funktionen?

A1: Det var mere tydeligt end den anden funktion. Men det var også fra starten af den funktion jeg synes var smartest fra starten af, så det kan være derfor jeg har bedre forståelse for hvorfor den virker.

Interviewer: Ja så det var simpelthen fordi du vidste at det var nyt i forhold til [Sonos].

A1: Ja jeg synes det var skide smart, og jeg glædede mig til at se hvad det var.

Interviewer: Ja okay.

Interviewer: Var der i perioder, når i brugte den her funktion, hvor den ikke opførte sig som forventet.

A1: Ja altså der var det med hullerne i musikken, de 15 sekunder, men ellers har der ikke været noget. Jo altså også når jeg gik i seng og jeg forventede det slukkede, men ellers ikke.

Interviewer: Hvad synes I om den funktion? Var den brugbar eller overflødig at have i f.eks. jeres Sonos-system?

A1: Den var skide smart. Den var tæt på uundværlig.

A2: Ja den var rigtig smart.

Interviewer: Og der kunne du også have tænkt dig at der var flere brugere?

A2: Ja.

Interviewer: Sådan som den virker, så måler den hvor tæt man er på den nye zone i forhold til den zone man kommer fra før den tænder. Så det er afstanden der afgør hvor den ene tænder og den anden slukker.

A1: Okay.

Interviewer: Men du [Interviewer spørger A1] synes det var smartere at have en tidsforsinkelse på, altså så det ikke udelukkende var baseret på afstand, eller hvordan?

A1: Ja. Og så ville jeg netop også have brugt den når jeg gjorde rent, så der ikke kom huller.

A2: Ja det ville have været lækkert med overlap i systemet.

Interviewer: Det var faktisk det jeg havde.

A1: Det var dejligt nemt.

Interviewer: Mange tak fordi I ville afsætte Jeres tid til at deltage i vores test af vores system.

A1: Det var så lidt, det var rigtig spændende at være med til at prøve sådan noget her.

APPENDIX C

Interview with Family B

Interviewer: Først vil jeg gerne høre dig hvad du overordnet synes om systemet med dine egne ord? Fungerede det som du forventede?

B1: Nu er jeg selv vant til at bruge mit eget system sådan til dagligt, og jeg synes det har været sjovt at prøve et lignende system, men som alligevel er så forskelligt fra det jeg normalt bruger. Jeg havde store forventninger til de nye funktionaliterer efter de blev præsenteret for mig, og jeg vil sige at jeg synes de har levet op til det som jeg forventede.

Interviewer: Var der noget som du synes manglede i forhold til dit eget AirPlay-baserede system?

B1: Altså jeg synes det var fedt at jeg kunne bruge mine playlists fra iTunes som jeg normalt bruger i stedet for at skulle lave nye. Det synes jeg. Der var dog det med at jeg flere gange oplevede at sangkøen blev fuld - også selvom at jeg havde hørt musikken.

Interviewer: Okay, så du ville gerne have ønsket at der kunne være flere sange i sangkøen?

B1: Det var ikke noget egentligt problem fordi jeg kunne jo gå ind og fjerne de sange jeg havde hørt, det skulle den måske gøre automatisk.

Interviewer: Ja okay. Ud over den begrænsede sangkø, var der ellers andet som du synes manglede i forhold til det system du benytter til dagligt?

B1: Nej ikke umiddelbart. Jo det skulle måske være understøttelse af netradio, som jeg hører engang imellem men ellers ikke.

Interviewer: Okay, det er noteret. Var der noget som du ville ønske havde været anderledes i systemet?

B1: Nej jeg vil sige at jeg er positivt overrasket over jeres system. Jeg synes at ideen med at kunne bevæge sig rundt med musikken i huset, ja den er rigtig god. Jeg er lidt træt af jeg ikke har den mulighed mere.

Interviewer: I forhold til brugen af systemet, hvor meget har I cirka anvendt systemet? Hvor mange timer om ugen eller om dagen?

B1: Både for min kones og mit vedkommende vil jeg sige at vi har brugt det mest inden vi tager på arbejde og så når vi er færdige med aftensmaden, ja jeg vil nok tro vi har benyttet det et par timer ca. fire-fem gange om ugen.

Interviewer: Har I brugt systemet på andre tidspunkter, f.eks. når I har haft gæster?

B1: Ja, jeg har vist systemet frem for et par bekendte i forbindelse med møder jeg har afholdt i hjemmet. Ja faktisk har jeg også brugt systemet en del når jeg har arbejdet hjemmefra. Der har det været rart at musikken har spillet i det rum jeg befinder mig i, og at det er det samme musik jeg kunne tage med mig.

Interviewer: Ja okay.

Interviewer: Har du eller I oplevet problemer i forbindelse med systemet i perioden?

B1: Nej ikke rigtigt. Altså der har været et par småting, men det har jeg selv kunne rette op på ved at bruge panikknappen i [android-applikationen].

Interviewer: Hvad med selve netværkssignalet? Har du oplevet udfald enten i forbindelse med afspilning af musik eller i forbindelse med at bruge telefonen til at styre musikken?

B1: Nej overhovedet ikke. Der har været lidt med at hvis man stod mellem to rum, så var musikken en anelse usynkroniseret, men det lærte jeg hurtigt at jeg kunne ordne ved at skifte til næste nummer og så tilbage igen. Men forbindelsen har været helt fin.

Interviewer: Efter din opfattelse, hvordan adskilte vores system sig fra dit AirPlay-system?

B1: Jeg synes det er utroligt at I kan bruge den hardware jeg selv har haft hele tiden, og så bygge et lag ovenpå som gør at jeg kan alt det nye her. Det adskiller sig markant i det med at jeg kan få musikken til at følge med rundt, og så også bare det at telefonen ved hvilket rum den selv er i, det synes jeg sgu er smart.

Interviewer: Vi har implementeret to funktioner som vores studie handler om. Den ene hedder location og den anden hedder movement. I location, som er den funktion som aktiveres øverst til venstre [Interviewer peger på ikonet] på den lille pil, så fjernbetjeningen skifter i forhold til det rum man befinder sig i.

B1: Ja?

Interviewer: Vil du sætte ord på hvordan du opfattede at den fungerede?

B1: Jamen den fungerer sådan at når den er aktiveret, jamen så vælger den den nærmeste [zone] i forhold til hvor jeg befinder mig. Det gjorde det hele lidt nemmere.

Interviewer: Er det en funktion I har anvendt?

B1: Ja bestemt. I stedet for at skulle vælge det rum jeg er i, som jeg jo normalt skal, jamen så vidste telefonen det allerede, og så kunne jeg bare styre musikken.

Interviewer: Hvor meget har I brugt funktionen?

B1: Jeg vil skyde på at jeg har brugt den.. Jamen jeg har brugt den det meste af tiden. Jeg har faktisk ikke rigtig haft behov for at styre et andet rum end det jeg nu var i.

Interviewer: Var I i tvivl om hvad funktionen gjorde og hvorfor den gjorde som den gjorde?

B1: Jeg synes at den var rigtig intuitiv at bruge. Den vælger jo bare det rum der er tættest på. Selvom det virker simpelt så er det jo utrolig brugbart.

Interviewer: Oplevede I under testperioden at den her funktion ikke opførte sig som den skulle?

B1: Nej ikke umiddelbart. Altså vi oplevede at [systemet] opførte sig underligt efter at jeg havde redigeret i mit iTunes bibliotek. Det var noget med nogle spillelister og lidt. Men efter jeg genstartede [brugte panikknappen på serverapplikationen] programmet på computeren så virkede det helt fint igen.

Interviewer: Hvad synes I om funktionen? Er den brugbar eller overflødig i forhold til det I er vant til?

B1: Som jeg sagde tidligere, så synes jeg at en relativt simpel funktion som den her faktisk er utrolig god. Det med at jeg ikke behøver at finde det rum jeg nu skal spille musikken i, jamen det gør det meget let at bruge selve mobiltelefonen til at styre musikken nu hvor man lige er.

Interviewer: Den anden funktion er movement, som er den funktion hvor musikken følger Jer rundt, altså Jeres bevægelser, som kunne aktiveres fra den her knap øverst til højre. [Interviewer peger på knappen]. Vil du beskrive med dine ord hvad den gør i systemet?

B1: Jamen jeg synes den fungerede rigtig godt. Selve funktionen virkede sådan at når jeg gik rundt i mit hus, så når jeg f.eks. arbejdede hjemme og skulle gå fra kontoret og ud i køkkenet, jamen så fulgte den musik med som jeg hørte inde på kontoret med ud i køkkenet. Jeg synes den fungerede rigtig godt.

Interviewer: Er det en funktion I har benyttet?

B1: Ja, bestemt. Vi har brugt den rigtig meget, dels fordi vi skulle teste systemet men også fordi vi synes den er smart og sjov at bruge. Det med at musikken følger efter er jo dejligt når man går herhjemme i sine egne tanker.

Interviewer: Var I i tvivl om hvordan selve funktionen skulle bruges?

B1: Altså ikke som sådan. Selvfølgelig skal man lige vænne sig til hvordan den fungerer første gang man bruger den, men jeg synes ikke vi var i tvivl om hvordan den skulle bruges.

Interviewer: Så I var på intet tidspunkt i tvivl om hvorfor den opførte sig som den gjorde?

B1: Jeg synes hurtigt at man fik en god forståelse for hvordan det virkede. Altså den starter jo bare musikken inden man kommer ind i næste rum, og så slukker den efter én. Det er jo dejlig nemt.

Interviewer: Var der perioder hvor systemet ikke opførte sig som forventet?

B1: Altså i det store hele, så fungerede det godt. Der var enkelte tilfælde, som jeg sagde tidligere, hvor musikken ikke var helt synkron, men det lærte vi hurtigt at fikse ved at skifte mellem numrene.

Interviewer: Okay, så det at skifte nummer i systemet var nok til at få zonerne synkrone igen?

B1: Ja det virkede i hvert fald sådan.

Interviewer: Hvad synes i om funktionen? Var den brugbar eller overflødig i forhold til det AirPlay system I normalt bruger?

B1: Jeg synes bestemt den var brugbar. Det bliver næsten trist at skulle undvære den nu hvor jeg har haft adgang til funktionen i et par uger. Jeg synes helt bestemt det har potentiale.

Interviewer: Det var godt.

Interviewer: Jamen det var det hele, mange tak fordi dig og din familie gad at hjælpe med at teste vores system.

B1: Selv tak.